

SCRUM BODY OF KNOWLEDGE (SBOK® LEITFADEN)

Vierte Edition (4th)

Beinhaltet zwei Kapitel zur Skalierung von
Scrum für große Projekte und für das Unternehmen

Ein umfassender Leitfaden zur Projektdurchführung mittels Scrum



SCRUM BODY OF KNOWLEDGE (*SBOK*[®] Leitfaden)

Vierte Edition (4th)

**Beinhaltet zwei Kapitel zur Skalierung von Scrum für
große Projekte und für das Unternehmen**

Ein umfassender Leitfaden zur Projektdurchführung mittels Scrum

©2022 SCRUMstudy™ – Eine Marke der VMEdU, Inc. Alle Rechte vorbehalten.

Library of Congress Cataloging-in-Publication Data

A guide to the Scrum Body of Knowledge (SBOK®), -- Englische Ausgabe von 2022

Beinhaltet bibliographische Referenzen und Indizes.

ISBN: 978-0-9899252-0-4

1. Scrum Methodik. I. SCRUMstudy. II. SBOK®

2013950625

ISBN: 978-0-9899252-0-4

Herausgegeben von:

SCRUMstudy – Eine Marke der VMEdU, Inc.

12725 W. Indian School Road, Suite F-112

Telefon: +1-480-882-0706

Fax: +1-240-238-2987

Email: sbok@scrumstudy.com

Internetseite: www.scrumstudy.com

„SBOK“, das SCRUMstudy Logo, „SFC“, „SDC“, „SMC“, „SAMC“, „SPOC“, und „ESMC“ sind eingetragene Handelsmarken von SCRUMstudy (Eine Marke der VMEdU, Inc.). Für eine umfassende Liste der SCRUMstudy Marken kontaktieren Sie bitte die SCRUMstudy Rechtsabteilung.

Der Scrum Body of Knowledge (SBOK®) wird ausschließlich zu Ausbildungszwecken ausgegeben. SCRUMstudy oder VMEdU, Inc. geben keine Garantie, dass er für jegliche andere Anwendungen nutzbar ist und beinhaltet keine ausdrückliche oder unterstellte Garantie jeglicher Art. Außerdem übernehmen sie keine Verantwortung für Fehler und Auslassungen. Keine Art der Verantwortung wird für einen nebensächlichen Schaden oder einen Folgeschaden, in Verbindung mit oder wegen einer Benutzung der vorliegenden Informationen, vorausgesetzt.

SCRUMstudy heißt Korrekturen und Kommentare bezüglich des Buches willkommen. Sollten Sie drucktechnische, formatbedingte oder andere Fehler finden, würden wir uns freuen, wenn Sie uns darüber in Kenntnis setzen. Sie können die relevante Seite des Buches kopieren, den Fehler markieren und uns an die oben erwähnte Adresse oder an folgende Emailadresse: support@scrumstudy.com or sbok@scrumstudy.com zusenden.

Kein Teil dieser Arbeit darf auf irgendeine Art elektronisch, manuell, durch Fotokopien, durch Aufnahmen oder durch jegliche Art der Informationsspeicherung oder Datenabfragungssystemen ohne vorherige Zustimmung des Verlegers reproduziert oder versendet werden.

10 9 8 7 6 5 4 3 2 1

Vorwort

Der *Scrum Body of Knowledge (SBOK®)* bietet Richtlinien für eine erfolgreiche Implementierung von Scrum – des populärsten Agile Produktentwicklungs- und Projektabwicklungsansatzes. Scrum wird im *SBOK®* als Rahmenstruktur, welche auf Portfolios, Programme oder Projekte jeder Größe anwendbar ist, definiert; Scrum kann in jeder Branche für Produkte, Dienstleistungen oder jedes andere zu liefernde Resultat angewendet werden.

Der *SBOK®* ist als Referenz und Ratgeber sowohl für erfahrene Praktiker von Scrum und anderen Produkt- oder Dienstleistungs-Entwicklungsmethoden vorgesehen, als auch für Personen, die noch keine Vorerfahrung oder Vorwissen zu Scrum oder einer anderen Projektabwicklungsmethode haben. Diese Neuausgabe des *SBOK®* liefert zusätzliche Einblicke in optimale Scrum Praktiken, insbesondere im Bereich der Skalierung von Scrum. Es wurden dem *SBOK®* zwei Kapitel hinzugefügt, die sich speziell mit *der Skalierung von Scrum* für große Projekte (Kapitel 13) und von Scrum für das Unternehmen (Kapitel 14) auseinandersetzen. Während die Beliebtheit und Anwendung von Scrum stetig wächst und sich global weiterentwickelt, ist es unser Ziel, gezeigte Lehren und beste Praktiken als Teil des *SBOK®* zu teilen.

Der *SBOK®* zieht sein kombiniertes Wissen und seine Einblicke aus Tausenden von Projekten, quer durch verschiedene Organisationen und Industriezweige. Diese 3. Ausgabe fügt die kollektiven Beiträge von Experten in Scrum und Projekt-Management hinzu. Insbesondere das Feedback der globalen Scrum Community hat eine große Rolle bei der Identifizierung von Verbesserungen und Ergänzungen des *SBOK®* gespielt. Dies war wirklich eine gemeinschaftliche Leistung einer großen Anzahl von Experten und Praktikern aus verschiedenen Wissenszweigen.

Die breite Akzeptanz der *SBOK®* Rahmenstruktur standardisiert die Anwendung von Scrum durch Organisationen weltweit und trägt ganz erheblich dazu bei, die Investitionsrendite dieser Organisationen signifikant zu erhöhen. Zusätzlich regt sie dazu an, mehr Überlegungen anzustellen wie Scrum auf viele verschiedene Projekttypen angewendet werden kann. Das wird dann im Gegenzug zu einer Erweiterung und Anreicherung des Body of Knowledge führen und dementsprechend zukünftige Aktualisierungen dieses Leitfadens nach sich ziehen.

Obwohl der *SBOK®* eine umfassende Referenz und Ratgeber für Scrum ist, ist sein Inhalt so organisiert, dass die Referenzen einfach zu verstehen sind, unabhängig davon, ob der Leser über Vorwissen zu dem Thema verfügt oder nicht. Wir hoffen, dass jeder Leser ebenso viel lernt und Spaß mit diesem Leitfaden hat, wie wir beim Zusammentragen des gemeinschaftlichen Wissens und des Fachwissens hatten, das seinen Weg in dieses Buch gefunden hat.



Tridibesh Satpathy,
Chefautor, Scrum Body of Knowledge (*SBOK®*)

Inhaltsverzeichnis

1.	EINLEITUNG.....	1
1.1	Ein Überblick über Scrum	2
1.1.1	Ein kurzer geschichtlicher Überblick über Scrum	3
1.2	Warum Scrum anwenden?.....	4
1.2.1	Die Skalierbarkeit von Scrum	5
1.3	Der Zweck des SBOK®	6
1.4	Rahmenstruktur des SBOK®	8
1.4.1	Wie wende ich den SBOK® an?	9
1.4.2	Scrum Prinzipien	9
1.4.3	Scrum Aspekte	11
1.4.4	Scrum Prozesse	16
1.5	Scrum versus traditionelles Projektmanagement.....	22
2.	PRINZIPIEN	23
2.1	Einführung.....	23
2.2	Leitfaden für Kernrollen.....	24
2.3	Empirische Prozesssteuerung	24
2.3.1	Transparenz	24
2.3.2	Inspektion	26
2.3.3	Adaptierung	26
2.4	Selbstorganisation	29
2.4.1	Benefits der Selbstorganisation	29
2.5	Kollaboration	31
2.5.1	Vorteile der Kollaboration bei Scrum Projekten	31
2.5.2	Die Bedeutung von Colocation für Kollaboration	33
2.5.3	Zusammenarbeit in verteilten Teams	33
2.6	Wertbasierte Priorisierung	35
2.7	Time-boxing	37
2.7.1	Scrum Time-boxes	37
2.8	Iterative Entwicklung	40
2.9	Scrum versus zu traditionelles Projektmanagement	42

3.	ORGANISATION.....	43
3.1	Einleitung.....	43
3.2	Leitfaden für die Kernrollen.....	44
3.3	Scrum Projekt Rollen	44
3.3.1	Kernrollen	44
3.3.2	Nicht-zentrale Rollen	46
3.4	Product Owner.....	48
3.4.1	Stimme des Kunden (VOC).....	50
3.5	Scrum Master.....	50
3.6	Scrum Team	52
3.6.1	Wahl der Mitarbeiter	54
3.6.2	Die Größe des Scrum Teams	54
3.7	Scrum in Großen Projekten, Programmen und Portfolios	55
3.7.1	Wie Scrum in einem großen Projekt funktioniert.....	55
3.7.2	Zusätzliche Kernrollen in einem großen Projekt	56
3.7.3	Anwendung von Scrum auf Unternehmens-Ebene.....	58
3.7.4	Zusätzliche Kernrollen in Programmen und Portfolios	59
3.7.5	Beispiele von Projekten, Programmen und Portfolios.....	61
3.8	Zusammenfassung der Verantwortlichkeiten	63
3.9	Scrum versus traditionelles Projekt Management.....	64
3.10	Populäre Theorien aus dem Personalbereich und deren Relevanz in Scrum.....	64
3.10.1	Modell der Gruppendynamik nach Tuckman	64
3.10.2	Konfliktmanagement.....	65
3.10.3	Konfliktmanagement Techniken	66
3.10.4	Führungsstile	67
3.10.5	Hierarchie der Bedürfnisse Theorie nach Maslow	69
3.10.6	X-Theorie und Y-Theorie	70
4.	GESCHÄFTLICHE RECHTFERTIGUNG.....	71
4.1	Einleitung.....	71
4.2	Leitfaden für die Kernrollen.....	72
4.3	Wert-getriebene Lieferung	72
4.3.1	Verantwortlichkeiten des Product Owners in der Geschäftlichen Rechtfertigung	74

4.3.2	Verantwortlichkeiten der anderen Scrum Rollen in der Geschäftlichen Rechtfertigung	74
4.4	Die Wichtigkeit der Geschäftlichen Rechtfertigung	75
4.4.1	Faktoren um Geschäftliche Rechtfertigung zu bestimmen	75
4.4.2	Geschäftliche Rechtfertigung und der Lebenszyklus des Projekts	77
4.5	Techniken der Geschäftlichen Rechtfertigung	78
4.5.1	Einschätzung des Projekt Mehrwerts	79
4.5.2	Planung für Mehrwert	80
4.5.3	Relative Priorisierungsrangliste	83
4.5.4	Story Mapping	84
4.6	Kontinuierliche Wertrechtfertigung	84
4.6.1	Ertragswertanalyse	85
4.6.2	Kumulatives Flussdiagramm (CFD)	87
4.7	Bestätigen der Ertragsrealisierung	88
4.7.1	Prototypen, Simulationen und Vorführungen	88
4.8	Zusammenfassung der Verantwortlichkeiten	89
4.9	Scrum versus traditionelles Projekt Management	90
5.	QUALITÄT	91
5.1	Einleitung	91
5.2	Leitfaden für die Kernrollen	92
5.3	Qualitätsdefinition	92
5.3.1	Qualität und Umfang	92
5.3.2	Qualität und Geschäftswert	93
5.4	Abnahmekriterien und das Priorisierte Produkt Backlog	94
5.4.1	Abnahmekriterien schreiben	96
5.4.2	Definition von Ready	96
5.4.3	Definition von Done (oder Done Kriterien)	97
5.4.4	Minimale Done Kriterien	98
5.4.5	Akzeptieren oder Ablehnen eines Priorisierten Produkt Backlog Artikels	98
5.5	Qualitätsmanagement in Scrum	99
5.5.1	Qualitätsplanung	99
5.5.2	Qualitätskontrolle und Qualitätssicherung	101
5.5.3	Plan-Do-Check-Act (PDCA) Zyklus	102

5.6	Zusammenfassung der Verantwortlichkeiten	103
5.7	Scrum versus traditionelles Projekt Management	104
6.	Änderungen.....	105
6.1	Einleitung	105
6.2	Leitfaden für die Kernrollen	106
6.3	Überblick	106
6.3.1	Nicht genehmigte und genehmigte Change Requests	107
6.4	Änderungen in Scrum	109
6.4.1	Das Gleichgewicht an Flexibilität und Stabilität halten.....	109
6.4.2	Flexibilität erreichen.....	109
6.5	Änderungen integrieren	115
6.5.1	Änderungen an einem Sprint	115
6.6	Änderungen in Programmen und Portfolios	121
6.6.1	Im Portfolio	121
6.6.2	Im Programm.....	121
6.7	Zusammenfassung der Verantwortlichkeiten	123
6.8	Scrum versus traditionelles Projekt Management	124
7.	RISIKO	125
7.1	Einführung.....	125
7.2	Leitfaden für Kernrollen.....	126
7.3	Was ist Risiko?	126
7.3.1	Der Unterschied zwischen Risiken und Problemen	127
7.3.2	Risikoverhalten	127
7.4	Verfahrensweise beim Risikomanagement.....	128
7.4.1	Risikoidentifikation	128
7.4.2	Risikobewertung	130
7.4.3	Risikopriorisierung	133
7.4.4	Risikominderung.....	134
7.4.5	Risikokommunikation.....	135
7.5	Minimierung von Risiken mit Hilfe von Scrum	137
7.6	Risiken bei Portfolios und Programmen.....	138
7.6.1	Bei Portfolios	138

7.6.2	Bei Programmen.....	138
7.7	Zusammenfassung der Verantwortlichkeiten	140
7.8	Scrum versus traditionelles Projektmanagement.....	141
8.	INITIIERUNG	143
8.1	Projektvision Erstellen.....	147
8.1.1	Inputs.....	149
8.1.2	Hilfsmittel	151
8.1.3	Outputs.....	152
8.2	Scrum Master und Business Stakeholdern Identifizieren.....	154
8.2.1	Inputs.....	156
8.2.2	Hilfsmittel	157
8.2.3	Outputs.....	159
8.3	Scrum Team bilden.....	160
8.3.1	Inputs.....	162
8.3.2	Hilfsmittel	163
8.3.3	Outputs.....	165
8.4	Epic(s) entwickeln	167
8.4.1	Inputs.....	168
8.4.2	Hilfsmittel	171
8.4.3	Outputs.....	174
8.5	Priorisiertes Produkt Backlog Erstellen	176
8.5.1	Inputs.....	177
8.5.2	Hilfsmittel	179
8.5.3	Outputs.....	182
8.6	Release-Planung durchführen	185
8.6.1	Inputs.....	186
8.6.2	Hilfsmittel	188
8.6.3	Outputs.....	188
8.7	Datenflussdiagramm der Initiierungs-Phase	190
9.	PLANEN UND ABSCHÄTZEN	191
9.1	User Storys erstellen.....	195
9.1.1	Inputs.....	197

9.1.2	Hilfsmittel	198
9.1.3	Outputs	200
9.2	User Storys Abschätzen	202
9.2.1	Inputs	204
9.2.2	Hilfsmittel	205
9.2.3	Outputs	207
9.3	User Storys Zusagen	208
9.3.1	Inputs	210
9.3.2	Hilfsmittel	211
9.3.3	Outputs	211
9.4	Aufgaben Identifizieren	214
9.4.1	Inputs	216
9.4.2	Hilfsmittel	216
9.4.3	Outputs	218
9.5	Aufgaben Abschätzen	220
9.5.1	Inputs	221
9.5.2	Hilfsmittel	223
9.5.3	Outputs	224
9.6	Sprint Backlog aktualisieren	225
9.6.1	Inputs	227
9.6.2	Hilfsmittel	228
9.6.3	Outputs	229
9.7	Datenflussdiagramm der Planungs- und Abschätzungs-Phase	231
10.	IMPLEMENTIEREN	233
10.1.1	Inputs	239
10.1.2	Hilfsmittel	242
10.1.3	Outputs	243
10.2	Tägliches Standup durchführen	245
10.2.1	Inputs	246
10.2.2	Hilfsmittel	247
10.2.3	Outputs	249
10.3	Priorisiertes Produkt Backlog verfeinern	251

10.3.1	Inputs.....	252
10.3.2	Hilfsmittel.....	254
10.3.3	Outputs.....	255
10.4	Datenflussdiagramm der Implementierungs-Phase.....	256
11.	Review und Rückblick.....	257
11.1	Sprint Vorführen und Validieren.....	261
11.1.1	Inputs.....	263
11.1.2	Hilfsmittel.....	264
11.1.3	Outputs.....	265
11.2	Sprint Rückblick halten.....	267
11.2.1	Inputs.....	268
11.2.2	Hilfsmittel.....	269
11.2.3	Outputs.....	271
11.3	Datenflussdiagramm der Review- und Rückblicksphase.....	273
12.	RELEASE.....	275
12.1	Zu liefernde Ergebnisse ausliefern.....	278
12.1.1	Inputs.....	279
12.1.2	Hilfsmittel.....	281
12.1.3	Outputs.....	282
12.2	Release Rückblick halten.....	283
12.2.1	Inputs.....	284
12.2.2	Hilfsmittel.....	285
12.2.3	Outputs.....	286
12.3	Datenflussdiagramm der Release-Phase.....	287
13.	Scrum für große Projekte Skalieren.....	289
13.1	Auswirkungen eines Großen Projekts auf die grundlegenden Scrum Prozesse.....	291
13.1.1	Initiierung.....	291
13.1.2	Planen und Abschätzen.....	296
13.1.3	Implementieren.....	299
13.1.4	Review und Rückblick.....	302
13.1.5	Release.....	304
13.2	Zusätzliche Inputs und Outputs für große Projekte.....	306

13.2.1	Organisation für große Scrum Projekte*	306
13.2.2	Product Owner Kollaborationsplan*	306
13.2.3	Scrum Masters/Scrum Teams Kollaborationsplan*	307
13.2.4	Geteilte Ressourcen*	307
13.2.5	Team-Spezialisierung*	308
13.2.6	Umgebung und Umgebungsplan*	309
13.2.7	Release Vorbereitungsplan*	309
13.3	Zusätzliche Hilfsmittel für große Projekte	310
13.3.1	Kommunikationsplan für das große Projekt	310
13.3.2	Ressourcenplanung für das große Projekt*	310
13.3.3	Identifizierung der Umgebung(en)*	311
13.3.4	Zuordnung des Priorisierten Produkt Backlogs*	311
13.3.5	Scrum der Scrums Meeting*	312
13.3.6	Release Vorbereitungsmethoden*	313
13.3.7	Release Vorbereitungs-Sprint	313
13.3.8	Scrum Projekttool	314
14.	Scrum für das Unternehmen Skalieren	315
14.1	Auswirkungen von Programmen oder Portfolios auf die grundlegenden Scrum-Prozesse	317
14.1.1	Initiierung	317
14.1.2	Planung und Abschätzung	321
14.1.3	Implementierung	321
14.1.4	Review und Rückblick	323
14.1.5	Release	324
14.2	Zusätzliche Prozesse zur Skalierung von Scrum für das Unternehmen (Programm/Portfolio) ...	325
14.3	Programm- oder Portfolio-Teams Bilden/Aktualisieren	329
14.3.1	Inputs	330
14.3.2	Hilfsmittel	331
14.3.3	Outputs	331
14.4	Programm- oder Portfoliokomponenten Erstellen/Aktualisieren	333
14.4.1	Inputs	334
14.4.2	Hilfsmittel	335
14.4.3	Outputs	337

14.5	Scrum Guidance Body überprüfen und aktualisieren.....	339
14.5.1	Inputs.....	340
14.5.2	Hilfsmittel.....	340
14.5.3	Outputs.....	341
14.6	Priorisiertes Programm- oder Portfoliobacklog erstellen/verfeinern.....	343
14.6.1	Inputs.....	344
14.6.2	Industriestandards.....	346
14.6.3	Hilfsmittel.....	346
14.6.4	Outputs.....	348
14.7	Programm- oder Portfolio-Releases definieren/aktualisieren.....	349
14.7.1	Inputs.....	350
14.7.2	Lessons Learned des Scrum-Teams.....	351
14.7.3	Release-Planung.....	351
14.7.4	Hilfsmittel.....	351
14.7.5	Outputs.....	352
14.8	Programm- oder Portfolio-Release Rückblick halten.....	353
14.8.1	Inputs.....	354
14.8.2	Hilfsmittel.....	355
14.8.3	Outputs.....	355
	ANHANG A. EIN AGILE ÜBERBLICK.....	357
	ANHANG B. AUTOREN UND REZENSENTEN DES SBOK®.....	367
	REFERENZEN.....	371
	Glossar.....	373

Abbildungsverzeichnis

ABBILDUNG 1-1: SCRUM FLUSSDIAGRAM	2
ABBILDUNG 1-2: SBOK® RAHMENSTRUKTUR	8
ABBILDUNG 1-3: SCRUM PRINZIPIEN	10
ABBILDUNG 1-4: ORGANISATION IN SCRUM	13
ABBILDUNG 2-1: TRANSPARENZ BEI SCRUM	25
ABBILDUNG 2-2: INSPEKTION BEI SCRUM.....	26
ABBILDUNG 2-3: ADAPTIERUNG IN SCRUM	27
ABBILDUNG 2-4: PROBLEME BEIM WASSERFALL-PROJEKTMANAGEMENT	28
ABBILDUNG 2-5: ZIELE EINES SICH SELBST ORGANISIERENDEN TEAMS	30
ABBILDUNG 2-6: VORTEILE DER KOLLABORATION BEI EINEM SCRUM-PROJEKT	32
ABBILDUNG 2-7: WERTBASIERTE PRIORISIERUNG	36
ABBILDUNG 2-8: TIME-BOXING-SITUATIONEN BEI SCRUM MEETINGS	39
ABBILDUNG 2-9: SCRUM VS. TRADITIONELLES WASSERFALL MODELL	41
ABBILDUNG 3-1: SCRUM ROLLEN—ÜBERBLICK	46
ABBILDUNG 3-2: GESTELLTE FRAGEN WÄHREND EINES SCRUM DER SCRUMS MEETING	57
ABBILDUNG 3-3: WÜNSCHENSWERTE EIGENSCHAFTEN FÜR DIE SCRUM KERNROLLEN.....	54
ABBILDUNG 3-4: PHASEN DER GRUPPENENTWICKLUNG NACH TUCKMAN	65
ABBILDUNG 3-5: HIERARCHIE DER BEDÜRFNISSE THEORIE NACH MASLOW.....	69
ABBILDUNG 4-1: LIEFERUNG VON MEHRWERT IN SCRUM VERSUS TRADITIONELLEN PROJEKTEN.....	73
ABBILDUNG 4-2: HIERARCHIE DER VERANTWORTLICHKEIT DER GESCHÄFTLICHEN RECHTFERTIGUNG	74
ABBILDUNG 4-3: GESCHÄFTLICHE RECHTFERTIGUNG UND DER PROJEKT LEBENSZYKLUS	78
ABBILDUNG 4-4: KANO-ANALYSE	83
ABBILDUNG 4-5: BEISPIEL KUMULATIVES FLUSSDIAGRAMM (CFD)	87
ABBILDUNG 5-1: PROJEKT-INKREMENT FLUSSDIAGRAMM.....	95
ABBILDUNG 5-2: PDCA ZYKLUS IN SCRUM.....	102
ABBILDUNG 6-1: BEISPIEL GENEHMIGTER ÄNDERUNGSPROZESS	107
ABBILDUNG 6-2: AKTUALISIEREN DES PRIORISIERTEN PRODUKT BACKLOGS MIT GENEHMIGTEN ÄNDERUNGEN	108
ABBILDUNG 6-3: SCRUM MERKMALE UM FLEXIBILITÄT ZU ERREICHEN.....	110
ABBILDUNG 6-4: BEWEGGRÜNDE DER BUSINESS STAKEHOLDERN, ÄNDERUNGEN ANZUFragen.....	111
ABBILDUNG 6-5: BEWEGGRÜNDE DES SCRUM KERNTAMS, ÄNDERUNGEN ANZUFragen.....	112
ABBILDUNG 6-6: INTEGRATION VON ÄNDERUNG IN SCRUM.....	116
ABBILDUNG 6-7: AUSWIRKUNG DER ERWARTETEN ÄNDERUNGEN AUF DIE SPRINT-LÄNGE	117
ABBILDUNG 6-8: ÄNDERUNGEN IM PORTFOLIO UND PROGRAMM EINBINDEN.....	122
ABBILDUNG 7-1: BEISPIEL FÜR EINEN WAHRSCHEINLICHSBAUM.....	131
ABBILDUNG 7-2: BEISPIEL EINES PARENTO-DIAGRAMMS.....	131
ABBILDUNG 7-3: BEISPIEL EINES WAHRSCHEINLICHS- / AUSWIRKUNGSRASTERS	132
ABBILDUNG 7-4: ABLAUF DER RISIKOPRIORISIERUNG	134
ABBILDUNG 7-5: BEISPIEL EINES RISIKO-BURNDOWN-CHARTS.....	136
ABBILDUNG 7-6: HANDHABUNG VON RISIKEN IN PORTFOLIOS UND PROGRAMMEN	139
ABBILDUNG 8-1: ÜBERBLICK ÜBER DIE INITIIERUNGSPHASE.....	146
ABBILDUNG 8-2: ÜBERBLICK ÜBER DIE INITIIERUNGSPHASE (KERNPUNKTE).....	146

ABBILDUNG 8-3: PROJEKTVISION ERSTELLEN—INPUTS, HILFSMITTEL UND OUTPUTS	147
ABBILDUNG 8-4: PROJEKTVISION ERSTELLEN—DATENFLUSSDIAGRAMM	148
ABBILDUNG 8-5: DER VORGANG DER LÜCKENANALYSE	152
ABBILDUNG 8-6: SCRUM MASTER UND BUSINESS BUSINESS STAKEHOLDERN IDENTIFIZIEREN—INPUTS, HILFSMITTEL UND OUTPUTS	154
ABBILDUNG 8-7: SCRUM MASTER UND BUSINESS STAKEHOLDERN IDENTIFIZIEREN—DATENFLUSSDIAGRAMM	155
ABBILDUNG 8-8: SCRUM TEAM BILDEN—INPUTS, HILFSMITTEL UND OUTPUTS.....	160
ABBILDUNG 8-9: SCRUM TEAM BILDEN—DATENFLUSSDIAGRAMM	161
ABBILDUNG 8-10: EPIC(S) ENTWICKELN—INPUTS, HILFSMITTEL UND OUTPUTS.....	167
ABBILDUNG 8-11: EPIC(S) ENTWICKELN-DATENFLUSSDIAGRAMM	168
ABBILDUNG 8-12: PRIORISIERTES PRODUKT BACKLOG ERSTELLEN—INPUTS, HILFSMITTEL UND OUTPUTS	176
ABBILDUNG 8-13: ERSTELLEN EINES PRIORISIERTEN PRODUKT BACKLOGS: DATENFLUSSDIAGRAMM	177
ABBILDUNG 8-14: RELEASE-PLANUNG DURCHFÜHREN-INPUTS, HILFSMITTEL UND OUTPUTS	185
ABBILDUNG 8-15: RELEASE-PLANUNG DURCHFÜHREN—DATENFLUSSDIAGRAMM.....	186
ABBILDUNG 8-16: INITIIERUNGSPHASE-DATENFLUSSDIAGRAMM	190
ABBILDUNG 9-1: PLANUNGS- UND ABSCHÄTZUNGSÜBERSICHT.....	193
ABBILDUNG 9-2: PLANUNGS- UND ABSCHÄTZUNGSÜBERSICHT (KERNPUNKTE)	194
ABBILDUNG 9-3: USER STORYS ERSTELLEN—INPUTS, HILFSMITTEL UND OUTPUTS.....	195
ABBILDUNG 9-4: USER STORYS ERSTELLEN—DATENFLUSSDIAGRAMM.....	196
ABBILDUNG 9-5: USER STORYS GENEHMIGEN, ABSCHÄTZEN UND ZUSAGEN—INPUTS, HILFSMITTEL UND OUTPUTS ..	202
ABBILDUNG 9-6:USER STORYS GENEHMIGEN, ABSCHÄTZEN UND ZUSAGEN—DATENFLUSSDIAGRAMM	203
ABBILDUNG 9-7: USER STORYS ZUSAGEN—INPUTS, HILFSMITTEL UND OUTPUTS	208
ABBILDUNG 9-8: USER STORYS ZUSAGEN—DATENFLUSSDIAGRAMM.....	209
ABBILDUNG 9-9: AUFGABEN ABSCHÄTZEN—INPUTS,HILFSMITTEL UND OUTPUTS	214
ABBILDUNG 9-10: AUFGABENSCHÄTZUNG—DATENFLUSSDIAGRAMM	215
ABBILDUNG 9-11: AUFGABENSCHÄTZUNG—DATENFLUSSDIAGRAMM	221
ABBILDUNG 9-12: SPRINT BACKLOG AKTUALISIEREN—INPUTS, HILFSMITTEL UND OUTPUTS	225
ABBILDUNG 9-13: SPRINT BACKLOG AKTUALISIEREN—DATENFLUSSDIAGRAMM.....	226
ABBILDUNG 9-14 PLANUNGS- UND SCHÄTZUNGSPHASE—DATENFLUSSDIAGRAMM.....	231
ABBILDUNG 10-1: IMPLEMENTIERUNGSÜBERSICHT	235
ABBILDUNG 10-2: IMPLEMENTIERUNGSÜBERSICHT (KERNPUNKTE).....	236
ABBILDUNG 10-3: ZU LIEFERNDE ERGEBNISSE ERZEUGEN—INPUTS, HILFSMITTEL UND OUTPUTS.....	237
ABBILDUNG 10-4: ZU LIEFERNDE ERGEBNISSE ERZEUGEN—DATENFLUSSDIAGRAM	238
ABBILDUNG 10-5: SCRUMBOARD	240
ABBILDUNG 10-6: TÄGLICHES STANDUP DURCHFÜHREN—INPUTS, HILFSMITTEL, AND OUTPUTS.....	245
ABBILDUNG 10-7: TÄGLICHES STANDUP DURCHFÜHREN—DATENFLUSSDIAGRAMM	246
ABBILDUNG 10-8: PRIORISIERTES PRODUKT BACKLOG VERFEINERN—INPUTS, HILFSMITTEL UND OUTPUTS	251
ABBILDUNG 10-9: PRIORISIERTES PRODUKT BACKLOG VERFEINERN—DATENFLUSSDIAGRAMM	252
ABBILDUNG 10-10: IMPLEMENTIERUNGSPHASE—DATENFLUSSDIAGRAMM	256
ABBILDUNG 11-1: ÜBERBLICK DER REVIEW- UND RÜCKBLICKSPHASE.....	259
ABBILDUNG 11-2: ÜBERBLICK DER REVIEW UND RÜCKBLICKSPHASE (KERNPUNKTE)	260
ABBILDUNG 11-3: ÜBERSICHT UND RÜCKBLICKSPHASE—DATENFLUSSDIAGRAMM	273
ABBILDUNG 12-1: RELEASE ÜBERSICHT.....	276
ABBILDUNG 12-2: RELEASE ÜBERSICHT (KERNPUNKTE)	277

ABBILDUNG 12-3: ZU LIEFERNDE ERGEBNISSE AUSLIEFERN—INPUTS, HILFSMITTEL UND OUTPUTS.....	278
ABBILDUNG 12-4: ZU LIEFERNDE ERGEBNISSE AUSLIEFERN – DATENFLUSSDIAGRAMM	279
ABBILDUNG 12-5: RELEASE RÜCKBLICK HALTEN—INPUTS, HILFSMITTEL UND OUTPUTS	283
ABBILDUNG 12-6: RELEASE RÜCKBLICK HALTEN—DATENFLUSSDIAGRAMM.....	284
ABBILDUNG 12-7: RELEASEPHASE—DATENFLUSSDIAGRAMM	287
ABBILDUNG 14-1: SKALIERUNG VON SCRUM FÜR DAS UNTERNEHMEN.....	328
ABBILDUNG 14-2: PROGRAMM ODER PORTFOLIO TEAMS BILDEN/AKTUALISIEREN—INPUTS, HILFSMITTEL, UND OUTPUTS	329
ABBILDUNG 14-3: PROGRAMM ODER PORTFOLIO KOMPONENTEN ERSTELLEN/AKTUALISIEREN—INPUTS, HILFSMITTEL, UND OUTPUTS	333
ABBILDUNG 14-4: SCRUM DER SCRUM DER SCRUMS (SoSOS) MEETING	336
ABBILDUNG 14-5: SCRUM GUIDANCE BODY ÜBERPRÜFEN UND AKTUALISIEREN —INPUTS, HILFSMITTEL, UND OUTPUTS	339
ABBILDUNG 14-6: PRIORISIERTES PROGRAMM- ODER PORTFOLIOBACKLOG ERSTELLEN/VERFEINERN —INPUTS, HILFSMITTEL, UND OUTPUTS	343
ABBILDUNG 14-7: PRIORISIERTES PROGRAMM- ODER PORTFOLIORELEASES DEFINIEREN/AKTUALISIEREN —INPUTS, HILFSMITTEL, UND OUTPUTS	349
ABBILDUNG 14-8: PRIORISIERTES PROGRAMM- ODER PORTFOLIORELEASES DEFINIEREN/AKTUALISIEREN —INPUTS, HILFSMITTEL, UND OUTPUTS	353

Tabellenverzeichnis

TABELLE 1-1: ÜBERSICHT ÜBER SCRUM PROZESSE	16
TABELLE 1-3: SCRUM VERSUS TRADITIONELLES PROJEKTMANAGEMENT	22
TABELLE 3-1: VERANTWORTLICHKEITEN DES PRODUCT OWNERS IN SCRUM PROZESSEN	49
TABELLE 3-2: VERANTWORTLICHKEITEN DES SCRUM MASTERS IN SCRUM PROZESSEN	51
TABELLE 3-3:.....	57
TABELLE 3-4: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN IN BEZUG ZUR ORGANISATION.....	63
TABELLE 4-1: ERTRAGSWERT FORMELN	85
TABELLE 4-2: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN IN BEZUG ZUR GESCHÄFTLICHEN RECHTFERTIGUNG	89
TABELLE 5-1: STUFEN DER ABNAHMEKRITERIEN	98
TABELLE 5-2: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH QUALITÄT	103
TABELLE 6-1: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN.....	123
TABELLE 7-1: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN IM BEZUG AUF RISIKEN	140
TABELLE 13-1: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN.....	291
TABELLE 13-2: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN.....	296
TABELLE 13-3: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN.....	300
TABELLE 13-4: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN.....	302
TABELLE 13-5: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN.....	304
TABELLE 14-1: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN.....	318
TABELLE 14-2: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN.....	321
TABELLE 14-3: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN.....	323
TABELLE 14-4: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN.....	324

1. EINLEITUNG

Das Buch *Scrum Body of Knowledge (SBOK®)* ist ein Leitfaden zur erfolgreichen Implementierung von Scrum, der populärsten Agile Projekt Management und Produkt Entwicklungs Ansatz. Es beinhaltet einen umfassenden Rahmen der Prinzipien, Aspekte und Prozesse von Scrum.

Scrum, wie im *SBOK®* definiert, ist auf Folgendes anwendbar:

- Portfolios, Programme und/oder Projekte in jeder Branche
- Produkte, Dienstleistungen oder jegliche an Business Stakeholdern zu liefernde Ergebnisse
- Projekte jeglicher Größe oder Komplexität

Der Begriff "Produkt" in dieser Wissenssammlung kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann auf jedes Projekt in jeder Branche effektiv angewendet werden — von kleinen Projekten oder Teams mit lediglich sechs Team-Mitgliedern bis hin zu großen, komplexen Projekten mit mehreren Hundert Team-Mitgliedern.

Das erste Kapitel beschreibt den Zweck und Rahmen des *SBOK®* und bietet eine Einführung in die Schlüsselkonzepte von Scrum. Es enthält eine Zusammenfassung von Scrum Prinzipien, Scrum Aspekten und Scrum Prozessen. Das zweite Kapitel vertieft die sechs Scrum Prinzipien, welche die Grundlage für die Scrum Rahmenstruktur bilden. Vom dritten bis zum siebten Kapitel werden die fünf Scrum Aspekte ausgearbeitet, welche bei allen Projekten immer angesprochen werden müssen: Organisation, geschäftliche Rechtfertigung, Qualität, Änderungen und Risiko. Kapitel acht bis zwölf decken die neunzehn Scrum Prozesse ab, welche bei einem Scrum Projekt zum Einsatz kommen. Diese Prozesse sind Teil der fünf Scrum Phasen: Initiierung, Planung und Abschätzung, Implementierung, Review und Rückblick, und Freigabe. Diese Phasen beschreiben im Detail die Inputs und Outputs, die mit jedem Prozess in Verbindung stehen, sowie die verschiedenen Hilfsmittel die jeweils benutzt werden können. Einige Inputs, Hilfsmittel und Outputs sind vorgeschrieben. Diese sind als solche gekennzeichnet, während andere optional sind, abhängig vom spezifischen Projekt, den Anforderungen der jeweiligen Organisation und/oder deren Richtlinien, wie im Scrum Guidance Body (SGB) festgehalten. Kapitel 13 und 14 wurden dem *SBOK®* in dieser Ausgabe hinzugefügt. Die beiden neuen Kapitel geben eine Anleitung zur Anwendung von Scrum für große Projekte bzw. die Skalierung von Scrum für das Unternehmen.

Die folgenden Abschnitte werden in diesem Kapitel besprochen:

1.1 Ein Überblick über Scrum

1.2 Warum Scrum anwenden?

1.3 Der Zweck des *SBOK®*

1.4 Rahmenstruktur des *SBOK®*

1.5 Scrum versus traditionelles Projekt Management

1.1 Ein Überblick über Scrum

Ein Scrum Projekt beinhaltet einen kooperativen Aufwand, um ein neues Produkt, eine Dienstleistung oder ein anderes Resultat zu kreieren, wie in der Projektvisionserklärung definiert. Projekte unterliegen Beschränkungen hinsichtlich Zeit, Kosten, Umfang, Qualität, Ressourcen, organisatorischen Möglichkeiten und anderen Limitierungen, die es schwierig machen zu planen, auszuführen, zu managen und schlussendlich erfolgreich zu sein. Allerdings bietet eine erfolgreiche Implementierung der Resultate eines abgeschlossenen Projekts einen signifikanten geschäftlichen Nutzen für eine Organisation. Deshalb ist es für Organisationen wichtig, die angemessene Projekt Management Methodik auszuwählen und diese in die Praxis umzusetzen.

Scrum zählt zu den populärsten Agile Methodiken. Es ist eine lernfähige, iterative (schrittweise sich wiederholende), schnelle, flexible und effektive Methodik, gestaltet um einen signifikanten Wert sowohl schnell als auch über den gesamten Projektzeitraum hinweg zu liefern. Obwohl das Scrum-Rahmenwerk, wie es im SBOK® definiert ist, in erster Linie für die Durchführung von Projekten und die Erstellung von Produkten verwendet wird, kann Scrum auch für die kontinuierliche Wartung von Produkten und Dienstleistungen, die Verfolgung von Problemen und das Management von Änderungen eingesetzt werden.

Scrum stellt Transparenz in der Kommunikation her, kreiert eine Umgebung der kollektiven Verantwortung und sichert einen kontinuierlichen Fortschritt. Die Scrum Rahmenstruktur, wie im SBOK® definiert, ist so strukturiert, dass sie die Entwicklung eines Produkts oder einer Dienstleistung in jedweder Branche oder jeder Art von Projekt, unabhängig von seiner Komplexität, unterstützt.

Eine wesentliche Stärke von Scrum liegt in funktionsübergreifenden, selbst-organisierten und ermächtigten Teams, die ihre Arbeit in kurze, hoch-konzentrierte Arbeitszyklen, sogenannte Sprints, aufteilen. Bild 1-1 gibt einen Überblick über den Fluss eines solchen Scrum Projekts.

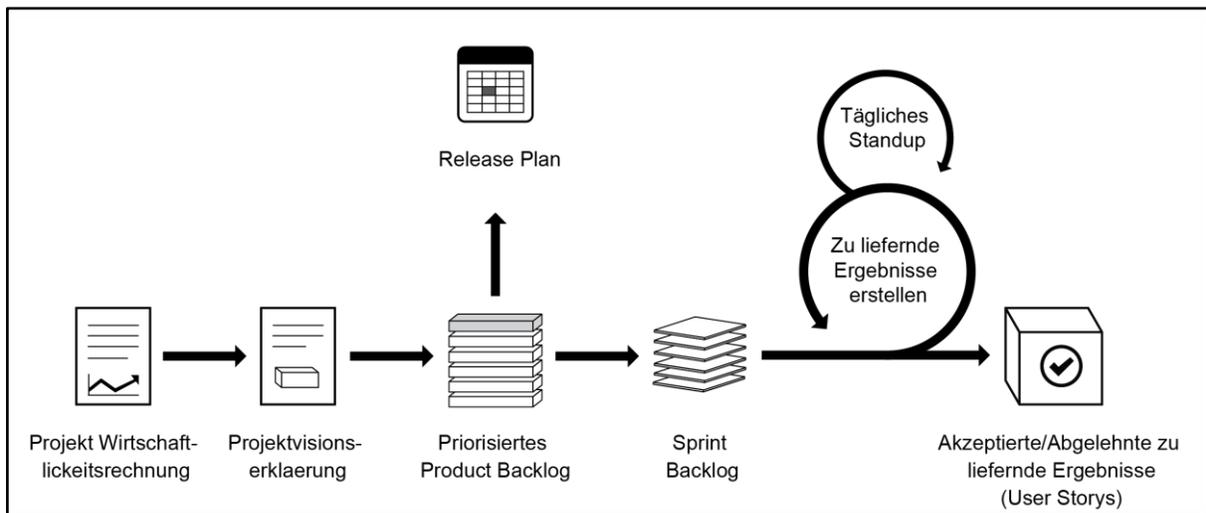


Abbildung 1-1: Scrum Flussdiagramm

Der Scrum Zyklus beginnt mit einem Business Stakeholder Meeting, währenddessen die Projektvision erstellt wird. Danach entwickelt der Product Owner das priorisierte Product Backlog, welches eine priorisierte Liste von wirtschaftlichen Anforderungen sowie Projektanforderungen enthält, die in der Form von User Stories geschrieben

sind. Jeder Sprint beginnt mit einem Sprint Planungsmeeting bei dem hochpriorisierte User Storys analysiert werden und entschieden wird, ob diese in den Sprint mit eingebunden werden sollen.

In der Regel dauert ein Sprint zwischen einer und sechs Wochen und beinhaltet die Arbeit des Scrum Teams lieferfähige zu liefernde Ergebnisse oder Produkt Inkremente zu erstellen. Während eines Sprints werden tägliche Standup Meetings durchgeführt, bei denen die Team Mitglieder den täglichen Fortschritt besprechen. Zum Ende des Sprints wird ein Sprint Review Meeting durchgeführt, bei dem der Product Owner und relevante Business Stakeholdern eine Vorführung der zu liefernden Ergebnisse bekommen. Der Product Owner akzeptiert die fertigen Ergebnisse nur dann, falls sie den vordefinierten Abnahmekriterien entsprechen. Der Sprint Zyklus endet mit einem Sprint Rückblick. Dabei wird ein Sprint Retrospektions-Meeting durchgeführt, bei dem das Team Wege diskutiert, wie Prozesse und die Leistung verbessert werden können, während sie zum nächsten Sprint übergehen.

1.1.1 Ein kurzer geschichtlicher Überblick über Scrum

Mitte der 80er Jahre haben Hirotaka Takeuchi und Ikujiro Nonaka eine flexible und allumfassende Produkt-Entwicklungsstrategie definiert, bei der das Entwicklungsteam zusammenarbeitet, um ein gemeinsames Ziel zu erreichen. Sie beschrieben einen innovativen Ansatz für die Produktentwicklung, welche sie als ganzheitlich bezeichneten oder auch „den Rugby-Ansatz“ nannten. Bei dieser Methode versucht das gesamte Team, die komplette Distanz als Einheit zu gehen und sich dabei „den Ball hin- und zurückzupassen“. Sie basierten diesen Ansatz auf Fallbeispielen aus der Produktion aus verschiedenen Industriezweigen. Takeuchi und Nonaka schlugen dabei vor, dass die Produktentwicklung nicht wie ein Staffellauf sein, sondern ähnlich dem Rugby Spiel ein Team zusammenarbeiten sollte, indem es sich den Ball hin- und zurückpasst, während es sich als Einheit bis zum Ende des Feldes vorarbeitet. Der Rugbygedanke beim Scrum-Konzept (bei dem sich eine Gruppe von „Spielern“ zusammenschließt, um ein Spiel erneut zu starten) wurde in diesem Artikel vorgestellt, um den Vorschlag des Autors zu beschreiben, dass die Produktentwicklung die „Vorwärtsbewegung des Scrums (zu deutsch; des Gedränges) entlang des Spielfeldes“ beinhalten sollte.

Seitdem haben mehrere Scrum-Praktiker, -Experten und -Autoren das Konzept und den Rahmen von Scrum auf der Grundlage von Best Practices weiter verfeinert. Ein wichtiger Meilenstein auf dem Weg zu Scrum war die Erstellung der ursprünglichen Version des SBOK® im Jahr 2013. Im Laufe der Zeit wurde der SBOK® Guide kontinuierlich verbessert, basierend auf den Bewertungen und dem Feedback von mehreren Tausend Scrum- und Agile-Praktikern, darunter mehr als 5000 Dozenten in mehr als 110 Ländern, die Scrum und Agile Praktiken unter Verwendung des SBOK® Guide-Frameworks unterrichten. Diese vierte Auflage ist das Ergebnis weiterer Verfeinerungen, die sicherstellen, dass der SBOK® Guide in einer sich ständig verändernden Welt weiterhin gültig und relevant bleibt.

Der SBOK® Guide ist heute der Industriestandard für Unternehmen und Praktiker, die an der Implementierung von Scrum oder Agile Praktiken interessiert sind. In den letzten Jahren hat Scrum an Popularität gewonnen und ist heute der bevorzugte Projektentwicklungsansatz für viele Organisationen weltweit.

Um seine Anwendung in multinationalen Umgebungen zu erleichtern, wurde der SBOK® Guide in mehrere Sprachen übersetzt, darunter Spanisch, Portugiesisch, Französisch, Italienisch, Arabisch, Chinesisch und Japanisch. Weitere Informationen zu den übersetzten Versionen finden Sie unter www.scrumstudy.com.

1.2 Warum Scrum anwenden?

Einige der Hauptvorteile beim Anwenden von Scrum in jeglichen Projekten sind:

1. **Anpassungsfähigkeit** — Die empirische Prozesssteuerung und die iterative (schrittweise sich wiederholende) Lieferung machen Projekte anpassungsfähig und ermöglichen Änderungen.
2. **Transparenz** — Alle Hilfsmittel zur Informations-Verbreitung wie zum Beispiel das Scrumboard und das Sprint Burndown Chart sind für alle verfügbar und führen zu einer offenen Arbeitsumgebung.
3. **Kontinuierliches Feedback** — Kontinuierliches Feedback ist durch die Prozesse *tägliches Standup Meeting durchführen* und *Sprint durchführen und validieren* gegeben.
4. **Kontinuierliche Verbesserung** — Durch den Prozess *Priorisiertes Produkt Backlog verfeinern* werden die zu liefernden Ergebnisse progressiv, von Sprint zu Sprint verbessert.
5. **Kontinuierliche Wertlieferung** — Iterative Prozesse ermöglichen die kontinuierliche Lieferung von Mehrwert durch den Prozess *zu liefernde Ergebnisse ausliefern*, so häufig wie der Kunde es möchte.
6. **Nachhaltiges Arbeitstempo** — Scrum Prozesse sind so gestaltet, dass die involvierten Personen in einem nachhaltigen Arbeitstempo arbeiten können; theoretisch für immer.
7. **Frühe Lieferung von hohem Wert** — Der *Priorisiertes Produkt Backlog erstellen* Prozess stellt sicher, dass die Kundenanforderungen mit dem höchsten Wert als erste erfüllt werden.
8. **Effiziente Entwicklungsprozesse** — Time-Boxing und Minimalisieren von unwichtigen Arbeitsschritten führen zu einem höheren Effizienzlevel.
9. **Motivation** — Die Prozesse *Tägliches Standup durchführen* und *Sprint Rückblick halten* führen zu einer höheren Motivation bei den Mitarbeitern.
10. **Schnellere Problemlösung** — Kollaboration und Zentralisierung von funktionsübergreifenden Teams, führen zu einer schnelleren Problemlösung.
11. **Effektivere zu liefernde Ergebnisse** — Der „Priorisiertes Produkt Backlog erstellen Prozess“ und reguläre Reviews nach Erstellung von zu liefernden Ergebnissen, sichern effektivere zu liefernde Ergebnisse an den Kunden.
12. **Der Kunde im Mittelpunkt** — Den Schwerpunkt auf geschäftliche Werte und ein kollaboratives Vorgehen mit den Business Stakeholdern zu legen, sichern ein kundenorientiertes Umfeld.

13. **Ein hohes Vertrauensumfeld** — Die Prozesse *Tägliche Standups* durchführen und *Sprint Rückblick halten* fördern Transparenz und Zusammenarbeit. Dies führt zu einem grossen Vertrauen im Arbeitsumfeld und stellt sicher, dass die Spannungen zwischen den Angestellten gering sind.
14. **Kollektive Verantwortung** — Der Prozess *User Storys Zusagen* erlaubt den Teammitgliedern kollektiv die Verantwortung fuer ein Projekt und für ihre Arbeit zu übernehmen und damit die Qualität zu verbessern.
15. **Hohe Geschwindigkeit** — Ein kollaborativer Rahmen ermöglicht es hoch ausgebildeten funktionsübergreifenden Teams, ihr volles Potenzial und damit eine hohe Arbeitsgeschwindigkeit zu erreichen.
16. **Innovative Umgebung** — Die Prozesse *Sprint Rückblick halten* und *Release Rückblick halten* formen eine Umgebung der Selbstprüfung, des Lernens und der Anpassungsfähigkeit, welche zu einer innovativen und kreativen Arbeitsumgebung führen.

1.2.1 Die Skalierbarkeit von Scrum

Um effektiv funktionieren zu können, sollte ein Scrum Team idealerweise zwischen sechs und zehn Mitgliedern bestehen. Diese Regel mag der Grund dafür sein, dass es eine falsche Vorstellung darüber gibt, dass Scrum nur in kleinen Projekten Anwendung finden kann. Scrum lässt sich sehr leicht für die effektive Anwendung in großen Projekten Programme und Portfolios skalieren. In Situationen, bei denen das Scrum Team zehn Personen übertrifft, können mehrere Scrum Teams geformt werden, die am Projekt arbeiten. Der logische Ansatz der Richtlinien und Prinzipien in dieser Rahmenstruktur kann auf das Managen von Projekten jeglicher Größe, die orts- und organisations-übergreifend sind, angewendet werden. Große Projekte können mehrere, parallel arbeitende Scrum Teams haben. In diesem Fall ist eine Synchronisation, eine Vereinfachung des Informationsflusses und eine Verbesserung der Kommunikation nötig. Große oder komplexe Projekte werden oft als Teile eines Programms oder Portfolios implementiert.

Weitere Informationen, die sich auf den Gebrauch von Scrum für große Projekte beziehen, sind in Kapitel 13 zu finden, die Skalierung von Scrum für Programme und Portfolios befinden sich in Kapitel 14.

1.3 Der Zweck des SBOK®

Das Scrum-Rahmenwerk hat sich als das bevorzugte Framework für die Projektdurchführung erwiesen, um durchgängig einen hohen Geschäftswert zu erzielen und die Investitionsrendite zu verbessern. Der Fokus von Scrum auf wertgesteuerte Lieferungen während eines Projekts hilft Scrum Teams, Ergebnisse so schnell wie möglich zu liefern.

Der SBOK® wurde als Leitfaden für Organisationen und für Projektmanagement Praktiker, die Scrum implementieren wollen, sowie für diejenigen, die Scrum schon jetzt anwenden, aber notwendige Verbesserungen bei den Prozessen durchführen möchten, entwickelt. Er basiert auf Erfahrungen aus Tausenden von Projekten aus einer Vielzahl von Organisationen und Industriezweigen. Die Beiträge vieler Scrum Experten und Projektmanagement Praktikern wurden in diese Entwicklung mit einbezogen.

Der SBOK® ist besonders wertvoll:

- für Scrum Kernteammitglieder, mit eingeschlossen:
 - Product Owner die die Scrum Rahmenstruktur völlig verstehen möchten, besonders die Teile, die Kunden- bzw. Business Stakeholdern-relevant sind, wie die geschäftliche Rechtfertigung, Qualität, Änderung und das mit Scrum Projekten verbundene Risiko.
 - Scrum Master, die ihre spezifische Rolle der Beaufsichtigung der Anwendung der Scrum Rahmenstruktur in Scrum Projekten erlernen möchten.
 - Scrum Teammitglieder, die sowohl Scrum Prozesse besser verstehen möchten und als auch die darin mit eingebundenen Hilfsmittel, die eventuell angewendet werden, um das Produkt oder die Dienstleistung des Projekts zu erstellen.
- Als ein umfassender Leitfaden für alle Scrum Praktiker, die an einem Scrum Projekt in egal welcher Organisation oder welchem Industriezweig, arbeiten.
- Als Referenzquelle für jede Person, die mit dem Scrum Kernteam interagiert, mit eingeschlossen (aber nicht beschränkt auf) Portfolio Product Owner, Portfolio Scrum Master, Programm Product Owner, Programm Scrum Master, Scrum Guidance Body, und Business Stakeholdernn (d.h. ein Sponsor, Kunde und Anwender).
- Als Handbuch für alle Personen, die kein Vorwissen oder generelles Wissen über die Scrum Rahmenstruktur haben, aber gerne Details über das Thema lernen möchten.

Außerdem ist der Inhalt des SBOK® eine Hilfe für Einzelpersonen, die sich auf eine der folgenden zertifizierten Abschlussprüfungen von Scrumstudy™ vorbereiten:

- Scrum Developer Certified (SDC®)
- Scrum Master Certified (SMC®)
- Scaled Scrum Master Certified (SSMC™)
- SCRUMstudy Agile Master Certified (SAMC™)

- Scrum Product Owner Certified (SPOC®)
- Scaled Scrum Product Owner Certified (SSPOC™)
- Expert Scrum Master Certified (ESMC™)

1.4 Rahmenstruktur des SBOK®

Der SBOK® ist in die folgenden drei Bereiche aufgeteilt:

1. **Prinzipien**, die im zweiten Kapitel besprochen werden, umfassen sechs Prinzipien, welche die Grundlage formen, auf der Scrum basiert.
2. **Aspekte**, die im dritten bis zum siebten Kapitel besprochen werden, beinhalten fünf Aspekte, die bei allen Scrum Projekten in Betracht gezogen werden sollten.
3. **Prozesse**, die in den Kapiteln acht bis zwölf beschrieben sind, beinhalten neunzehn grundlegende Scrum Prozesse und deren zugehörige Inputs, Hilfsmittel und Outputs. Kapitel 13 beinhaltet die zusätzlichen Inputs, Hilfsmittel und Outputs, die bei der Skalierung von Scrum für große Projekte benötigt werden, und Kapitel 14 beinhaltet die zusätzlichen Prozesse, die bei der Skalierung von Scrum für das Unternehmen benötigt werden.

Abbildung 1-2 zeigt die SBOK® Rahmenstruktur, und wie Prinzipien, Aspekte und Prozesse miteinander interagieren – alle drei sind von gleicher Bedeutung für das Verständnis und die Anwendung von Scrum.

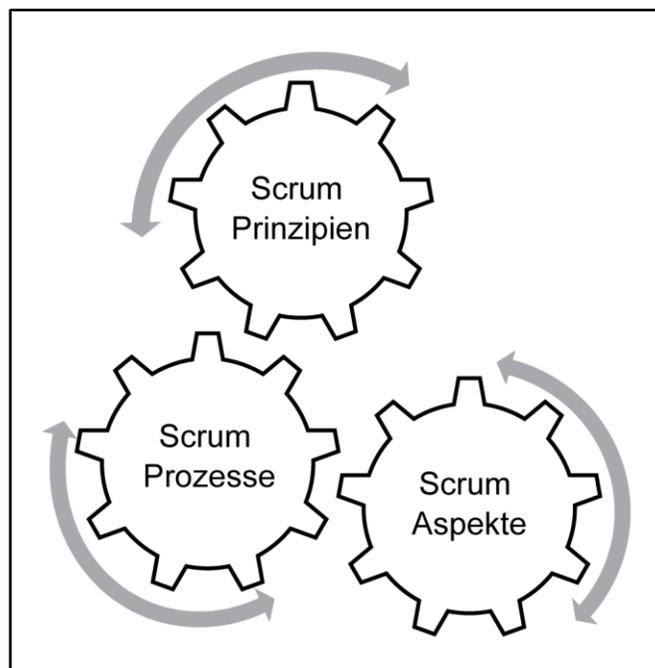


Abbildung 1-2: SBOK® Rahmenstruktur

1.4.1 Wie wende ich den SBOK® Leitfaden?

Der SBOK® kann als Referenz und Leitfaden sowohl von erfahrenen Praktikern von Scrum oder anderen Methoden zur Entwicklung von Produkten und Dienstleistungen angewendet werden, als auch von jemandem, der noch über keine Vorerfahrung oder generelles Wissen über die Scrum Projekt Management Methodik verfügt. Der Inhalt ist als einfache Referenz für die drei Scrum Kernteam Rollen organisiert: Scrum Master, Product Owner und Scrum Team.

Die Abschnitte, welche die sechs Scrum Prinzipien umfassen (Kapitel 2), sowie die fünf Scrum Aspekte (Kapitel 3 bis 7), beinhalten einen Leitfaden für die Kernrollen. Dieser Leitfaden bietet eine Anleitung hinsichtlich der Relevanz der jeweiligen Sektionen des Kapitels für die Scrum Kernteam Rollen.

Um die beste Anwendung der Scrum Rahmenstruktur zu vereinfachen, hat der SBOK® deutlich zwischen verbindlichen bzw. optionalen Inputs, Hilfsmitteln und Outputs unterschieden. Inputs, Hilfsmittel und Outputs mit einem Stern (*) sind verbindlich, oder werden als kritisch für den Erfolg des Projekts angesehen, während andere ohne Stern (*) optional sind. Es wird empfohlen, dass Personen, die sich gerade erst mit Scrum vertraut machen, sich hauptsächlich auf die verbindlichen Inputs, Hilfsmittel und Outputs konzentrieren, während erfahrenere Anwender vollständige Prozesskapitel lesen sollten um von den als Best Practices vorgeschlagenen Inputs, Hilfsmitteln und Outputs zu profitieren.

Scrum ist ein Rahmen und sollte auf keinen Fall als präskriptiv angesehen werden. D.h., es gibt Spielraum für Flexibilität in seiner Anwendung. Alle grundlegenden Scrum Prozesse, die im SBOK® ausführlich dargestellt werden (Kapitel 8 bis 12.), werden für jedes Scrum Projekt benötigt, werden aber basierend auf dem spezifischen Bedarf der jeweiligen Organisation, des Projekts, Produkts oder Teams angewendet. Zusätzliche Inputs, Hilfsmittel und Outputs kommen zur Anwendung, wenn Scrum für große Projekte skaliert wird (Kapitel 13). Zusätzliche Prozesse kommen zur Anwendung, wenn Scrum für das Unternehmen skaliert wird (Kapitel 14).

1.4.2 Scrum Prinzipien

Scrum Prinzipien sind Kernrichtlinien zur Anwendung der Scrum Rahmenstruktur und sollten als verbindliche Vorschriften bei allen Scrum Projekten angewendet werden. Die sechs Scrum Prinzipien, die im zweiten Kapitel vorgestellt werden, sind:

1. Empirische Prozesssteuerung
2. Selbstorganisation
3. Kollaboration
4. Wertbasierende Priorisierung
5. Time-Boxing
6. Iterative Entwicklung

Abbildung 1-3 beschreibt die sechs Scrum Prinzipien.

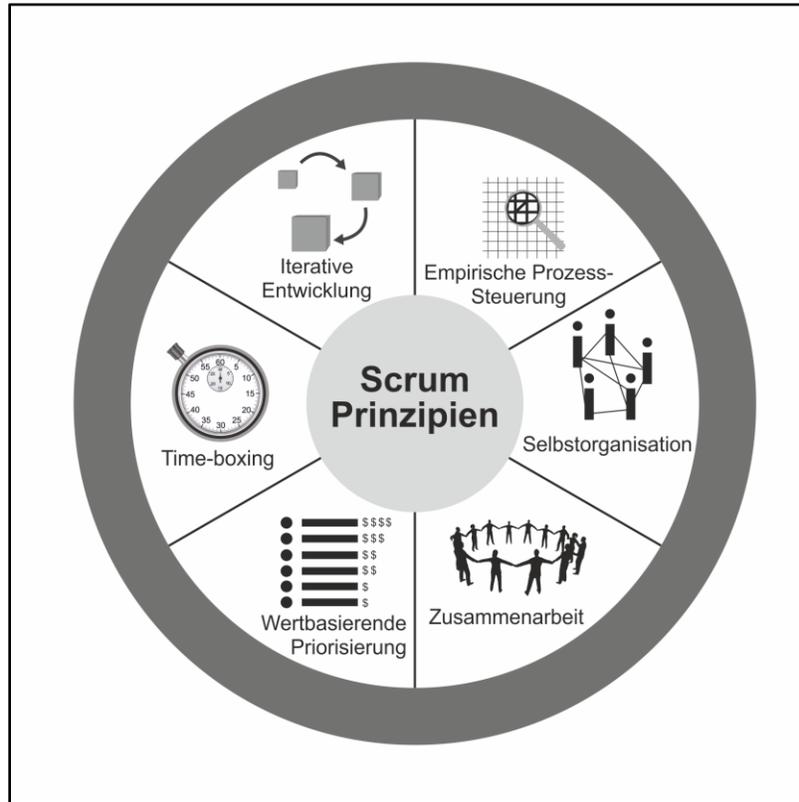


Abbildung 1-3: Scrum Prinzipien

Scrum Prinzipien können auf jegliche Arten von Projekten in allen Organisationen angewandt werden und müssen sichergestellt werden, um eine effektive Implementierung der Scrum Rahmenstruktur zu gewährleisten. Scrum Prinzipien sind nicht verhandelbar und müssen genauso wie im *SBOK®* spezifiziert angewendet werden. Indem die Prinzipien intakt gehalten und sie angemessen angewendet werden, wird Vertrauen in die Scrum Rahmenstruktur aufgebaut, um die Ziele eines Projekts zu erreichen. Die Scrum Aspekte und Prozesse hingegen können modifiziert werden, um Anforderungen an ein Projekt oder die der Organisation zu erreichen.

1. **Empirische Prozesssteuerung** — Dieses Prinzip hebt die Kernphilosophie von Scrum hervor, die auf den drei Hauptgedanken der Transparenz, Inspektion und Adaptierung basiert. Die empirische Prozesssteuerung unterstützt das Lernen durch Experimentieren, insbesondere wenn das Problem nicht genau definiert ist oder wenn es keine klaren Lösungen gibt.
2. **Selbstorganisation** — Dieses Prinzip legt den Fokus auf die heutigen Beschäftigten, welche, sofern sie selbstorganisiert arbeiten, einen signifikant höheren Wert liefern. Dies resultiert in einer erhöhten Akzeptanz durch das Team, einer gemeinsamen Verantwortung sowie einer wachstumsförderlichen, innovativen und kreativen Arbeitsumgebung.

3. **Zusammenarbeit** — Dieses Prinzip legt den Fokus auf die drei Kerndimensionen, die mit der kooperativen Arbeit verbunden sind: Bewusstsein, offene Aussprache und Übernahme von Verantwortung. Außerdem befürwortet es das Projektmanagement als einen gemeinsamen Wertschöpfungsprozess mit Teams, die nicht nur untereinander, sondern auch mit dem Kunden und anderen Business Stakeholdern zusammenarbeiten und interagieren, um als Ergebnis den höchstmöglichen Wert zu liefern.
4. **Wertbasierende Priorisierung** — Dieses Prinzip hebt den Scrum Fokus hervor, von Beginn des Projekts an und danach kontinuierlich den höchstmöglichen geschäftlichen Wert zu liefern.
5. **Time-Boxing** — Dieses Prinzip beschreibt, wie Zeit von Scrum als limitierende Einschränkung betrachtet wird und hilft dabei, die Projektplanung und Durchführung effektiv zu gestalten. Elemente, die bei Scrum einem festen Zeitfenster unterliegen, schließen Sprints, tägliche Standup Meetings, Sprint Planungsmeetings und Sprint Rückblick Meetings mit ein.
6. **Iterative Entwicklung** — Dieses Prinzip definiert die iterative Entwicklung und legt das Hauptmerkmal darauf, Änderungen besser zu managen und erklärt, wie man Produkte kreiert, die den Kunden zufriedenstellen und dessen Bedürfnisse erfüllen. Außerdem beschreibt es die Verantwortlichkeiten des Product Owners und die der Organisation in Verbindung mit iterativer Entwicklung.

1.4.3 Scrum Aspekte

Die Scrum Aspekte müssen durch das gesamte Scrum Projekt hindurch adressiert und gemanagt werden. Die fünf Scrum Aspekte, die in Kapitel drei bis sieben vorgestellt werden, sind:

1.4.3.1 Organisation

Definierte Rollen und Verantwortlichkeiten in einem Scrum Projekt zu verstehen ist sehr wichtig, um eine erfolgreiche Implementierung von Scrum sicherzustellen.

Scrum Rollen fallen in zwei breite Kategorien:

1. **Kernrollen** — Kernrollen sind die Rollen, welche unbedingt benötigt werden, um das Produkt oder die Dienstleistung des Projekts herzustellen. Einzelpersonen, denen Kernrollen zugeteilt werden, verpflichten sich dem Projekt komplett und sind schlussendlich verantwortlich für den Erfolg jeder einzelnen Iteration sowie des gesamten Projekts.

Diese Rollen sind:

- Der **Product Owner**, der dafür verantwortlich ist, den maximalen geschäftlichen Wert für das Projekt zu erreichen. Er oder sie ist auch dafür verantwortlich, die Kundenwünsche verständlich

zu machen und die geschäftliche Rechtfertigung für das Projekt auf dem aktuellen Stand zu halten. Der Product Owner repräsentiert die Stimme des Kunden.

- Der **Scrum Master** ist ein Moderator und Organisator und stellt sicher, dass das Scrum Team in einer Umgebung arbeitet, die der erfolgreichen Beendigung des Projekts förderlich ist. Der Scrum Master leitet das Team an, fördert es und lehrt Scrum Praktiken an alle, die mit dem Projekt verbunden sind; er räumt Hindernisse für das Team beiseite und stellt sicher, dass den Scrum Prozessen gefolgt wird.
 - Das **Scrum Team** ist eine Gruppe oder ein Team von Personen, die verantwortlich dafür sind, die vom Product Owner spezifizierten Anforderungen zu verstehen und die zu liefernden Ergebnisse für das Projekt zu erstellen.
- 2. Nicht-zentrale Rollen** — Nicht-zentrale Rollen sind solche Rollen, die nicht unbedingt für ein Scrum Projekt benötigt werden und können Teammitglieder einschließen, die ein gewisses Interesse am Projekt haben. Sie haben keine formelle Rolle im Projektteam, können aber mit dem Team in Verbindung stehen; allerdings sind sie nicht für den Erfolg des Projekts verantwortlich. Diese nicht-zentralen Rollen sollten in allen Scrum Projekten in Betracht gezogen werden.

Nicht-zentrale Rollen schließen folgende/folgendes mit ein:

- **Business Stakeholdern**, ein kollektiver Ausdruck für Kunden, Anwender und Sponsoren, tritt häufig mit dem Scrum Kernteam in Verbindung und nimmt während der gesamten Projektentwicklung Einfluss auf das Projekt. Es sind die Business Stakeholdern, für die durch das Projekt ein kollaborativer Nutzen hergestellt wird. Business Stakeholdern sind eine Teilmenge aller Business Stakeholdern eines Scrum Projekt – zu den Business Stakeholdern gehören alle Personen oder Gruppen, die vom Scrum Projekt betroffen sind, sowohl innerhalb als auch außerhalb der Organisation (z. B. alle Kernrollen und nicht-zentralen Rollen, Anbieter, interne Gruppen, Experten, usw.).
- **Der Scrum Guidance Body (SGB)** ist eine optionale Rolle, welche im Allgemeinen aus einem Dokumentensatz und/oder einer Expertengruppe besteht, die typischerweise damit beschäftigt ist, Ziele zu definieren, die in Verbindung mit Qualität, gültigen rechtlichen Bestimmungen, Sicherheit und anderen organisatorischen Schlüsselparametern stehen. Dieser SGB leitet die vom Product Owner, Scrum Master und Scrum Team durchgeführte Arbeit an.
- **Die Anbieter** sind externe Einzelpersonen oder Organisationen, die Produkte und Dienstleistungen bereitstellen, die nicht Teil der Kernkompetenzen der Projektorganisation sind.

Abbildung 1-4 zeigt die Scrum Organisationsstruktur.

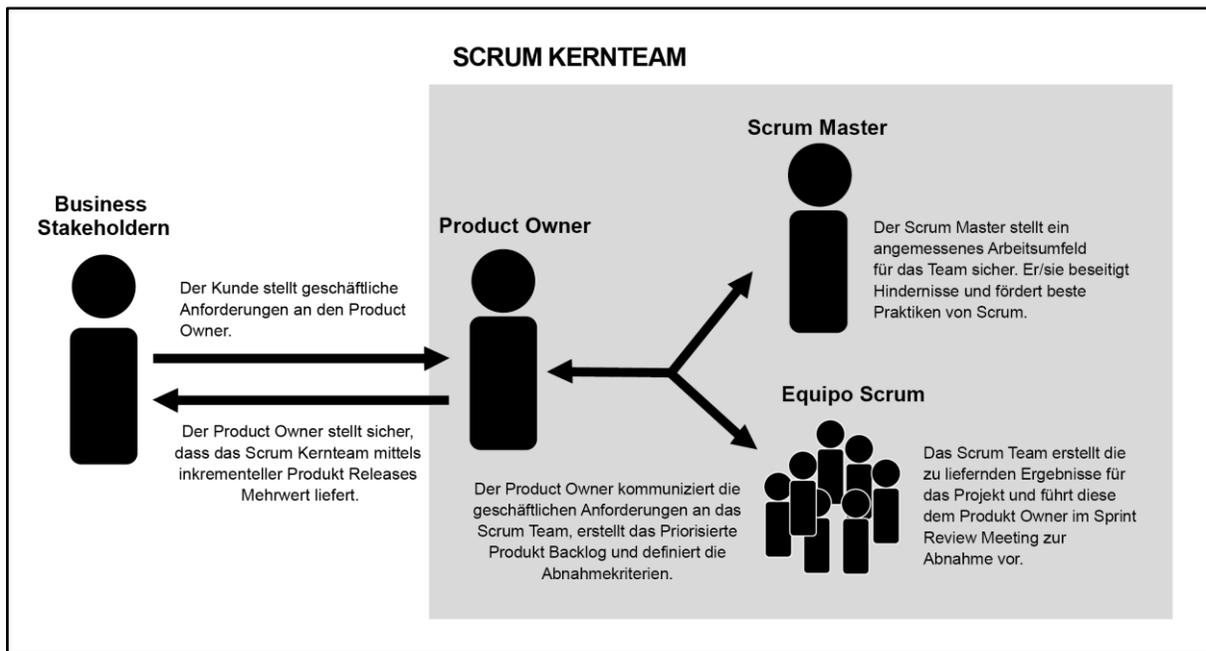


Abbildung 1-4: Organisation in Scrum

Die Organisationsaspekte von Scrum adressieren auch die Teamstrukturanforderungen, um Scrum in großen Projekten Programmen und Portfolios zu implementieren.

1.4.3.2 Geschäftliche Rechtfertigung

Vor dem Start eines Projekts ist es für eine Organisation wichtig, eine angemessene betriebswirtschaftliche Auswertung zu erstellen. Dies hilft den wichtigsten Entscheidungsträgern, die Machbarkeit eines Projekts sowie den wirtschaftlichen Bedarf für eine Änderung, ein neues Produkt oder eine Dienstleistung sowie die Rechtfertigung für den Start des Projekts, zu verstehen.

Die geschäftliche Rechtfertigung in Scrum basiert auf dem Konzept der wertgetriebenen Lieferung. Eine der Kerncharakteristiken bei allen Projekten ist die Ungewissheit der Resultate oder der Ergebnisse.

Unabhängig von der Größe oder Komplexität des Projekts ist es unmöglich, den Erfolg bei Beendigung eines Projekts zu garantieren.

In Betrachtung dieser Ungewissheit, einen Erfolg zu erreichen, versucht Scrum, Resultate so früh wie möglich, noch während das Projekt läuft, zu liefern. Diese frühe Lieferung von Resultaten, und damit Mehrwert, eröffnet die Möglichkeit für eine Reinvestition und bestätigt den interessierten Business Stakeholdern den Wert des Projekts.

Die Anpassungsfähigkeit von Scrum erlaubt es, die Projektziele und Prozesse zu verändern, falls die geschäftliche Rechtfertigung sich ändert. Hierbei ist es wichtig zu erwähnen, dass obwohl der Product Owner primär für die geschäftliche Rechtfertigung verantwortlich ist, dennoch andere Teammitglieder einen maßgeblichen Beitrag leisten.

1.4.3.3 Qualität

In Scrum wird Qualität als die Fähigkeit des fertigen Produkts oder der zu liefernden Ergebnisse, den Abnahmekriterien zu entsprechen und den geschäftlichen Wert, der vom Kunden erwartet wird, zu erreichen, definiert.

Um sicherzustellen, dass ein Projekt den Qualitätsanforderungen entspricht, eignet sich Scrum eine kontinuierliche Verbesserungsmethodik an, wobei das Team aus Erfahrung und dem Business Stakeholdern Engagement lernt, das priorisierte Produkt Backlog ständig auf dem neuesten Stand hinsichtlich etwaiger Änderungen der Anforderungen zu halten. Das priorisierte Produkt Backlog ist einfach niemals vollständig, außer das Projekt ist abgeschlossen oder wird abgebrochen. Jegliche Änderung der Anforderungen reflektiert eine Änderung im internen und externen geschäftlichen Umfeld und gestattet es dem Team, kontinuierlich zu arbeiten und sich an jene Anforderungen anzupassen, um sie zu erfüllen.

Der Umstand, dass Scrum verlangt, dass die Arbeit in Inkrementen und in Sprints vervollständigt wird, bedeutet, dass durch wiederholte Qualitätstests Fehler und Mängel früher entdeckt werden und nicht erst gegen Ende. Außerdem werden wichtige qualitätsbezogene Aufgaben (z.B. Entwicklung, Testen und Dokumentation) als Teil desselben Sprints vom selben Team fertiggestellt – dies sichert dass Qualität bei allen zu liefernden Ergebnissen als Teil eines Sprints inbegriffen ist. Solche als Teil eines Scrum Projekts zu liefernden Ergebnisse, die lieferfähig sind, werden auch als „Done“ bezeichnet.

Demnach optimiert die kontinuierliche Verbesserung mit wiederholten Tests die Wahrscheinlichkeit, das erwartete Qualitätsniveau in einem Scrum Projekt zu erreichen. Kontinuierliche Besprechungen zwischen dem Scrum Kernteam und den Business Stakeholdern (einschließlich Kunden und Anwendern) sowie die Lieferung von Inkrementen des Produkts zum Ende jeden Sprints stellen sicher, dass die Kluft zwischen den Kundenerwartungen an das Projekt und tatsächlich hergestellten zu liefernden Ergebnissen kontinuierlich reduziert wird.

Außerdem kann der Scrum Guidance Body Qualitäts-Richtlinien, die für die Organisation bei allen Scrum Projekten relevant sind, liefern.

1.4.3.4 Änderung

Jedes Projekt, gleichgültig, welche Methodik oder Rahmenstruktur verwendet wird, ist Änderungen unterworfen. Es ist unbedingt notwendig, dass Projekt-Teammitglieder verstehen, dass die Scrum Entwicklungsprozesse so entworfen sind, dass sie Änderungen positiv aufnehmen. Organisationen sollten den Nutzen, der aus einer

Änderung hervorgeht, maximieren und versuchen, jede negative Auswirkung durch sorgfältige Änderungs-Managementprozesse in Übereinstimmung mit den Prinzipien von Scrum zu minimieren.

Ein primäres Prinzip von Scrum ist seine Anerkennung, dass a) Business Stakeholdern (z.B. Kunden, Anwender und Sponsoren) während eines Projekts ihre Meinung über das, was sie benötigen und wollen, ändern (manchmal als „Anforderungsänderung“ bezeichnet) und b) dass es für die Business Stakeholdern sehr schwierig, wenn nicht unmöglich, ist alle Anforderungen während des Starts eines Projekts zu definieren.

Scrum Projekte heißen Änderung willkommen, indem sie kurze, iterative Sprints durchführen, welche ein Kundenfeedback zu jedem zu liefernden Ergebnis mit einschließen. Dies ermöglicht es dem Kunden, mit den Scrum Teammitgliedern regelmäßig zu interagieren, zu liefernde Ergebnisse einzusehen, sobald sie fertig sind, und Anforderungen gegebenenfalls früh im Sprint zu ändern.

Auch die Portfolio- oder Programm-Management Teams können, ihrem Level entsprechend, auf Scrum Projekt Änderungsanforderungen reagieren.

1.4.3.5 Risiko

Risiko wird definiert als ein ungewisses Ereignis oder eine Reihe von Ereignissen, die einen Effekt auf die Ziele eines Projekts haben können und zu einem Erfolg oder Misserfolg beitragen. Ein Risiko, oder Risiken, die mit hoher Wahrscheinlichkeit einen positiven Effekt auf das Projekt haben, werden als Gelegenheiten bezeichnet, während „Gefahren“ Risiken oder ein Risiko darstellen, die ein Projekt negativ beeinflussen könnten. Risikomanagement muss mit Eigeninitiative getätigt werden und ist ein iterativer Prozess. Er sollte mit dem Projektbeginn starten und wird durch den gesamten Projekt-Zyklus fortgesetzt. Der Risikomanagement Prozess sollte standardisierte Schritte haben, um sicherzustellen, dass ein Risiko oder Risiken identifiziert und eingeschätzt werden und eine angemessene Vorgehensweise festgelegt und entsprechend befolgt wird.

Ein Risiko, oder mehrere Risiken, sollte/n auf zwei Faktoren basierend identifiziert, eingeschätzt und angegangen werden: Die Wahrscheinlichkeit des Auftretens eines jeden Risikos und die möglichen Auswirkungen im Fall eines Auftretens. Risiken mit einem hohen Wert an Auftrittswahrscheinlichkeit und Auswirkung (festgelegt, indem beide Faktoren multipliziert werden), sollten vor denen mit einem niedrigeren Wert bearbeitet werden. Sobald ein Risiko identifiziert wurde, ist es generell wichtig zu verstehen, welchen wahrscheinlichen Ursprung und welchen potenziellen Effekt das Risiko hat, falls es auftreten wird.

1.4.4 Scrum Prozesse

Scrum Prozesse sprechen spezifische Aktivitäten und den Scrum Projektfluss an. Scrum Prozesse werden im Allgemeinen nicht sequenziell durchgeführt, sondern sind iterativer Natur und können sich durchaus überschneiden. Insgesamt gibt es neunzehn grundlegende Scrum Prozesse, die für alle Projekte anwendbar sind. Diese Prozesse sind in fünf Phasen eingeteilt und sind in den Kapiteln acht bis zwölf im SBOK® zu finden, wie in Tabelle 1-1 aufgezeigt.

Kapitel	Phase	Prozesse
8	Start	1. Projektvision erstellen 2. Scrum Master und Business Stakeholdern identifizieren 3. Scrum Team bilden 4. Epic(s) entwickeln 5. Priorisiertes Produkt Backlog erstellen 6. Release-Planung durchführen
9	Planung und Abschätzung	7. Anwender Storys erstellen 8. User Storys abschätzen 9. User Storys zusagen 10. Aufgaben <i>identifizieren</i> 11. Aufgaben abschätzen 12. Sprint Backlog aktualisieren
10	Implementierung	13. Zu liefernde Ergebnisse erzeugen 14. Tägliche Standups durchführen 15. Priorisiertes Produkt Backlog verfeinern
11	Rückblick und Retrospektion	16. Sprint durchführen und validieren 17. Sprint Rückblick halten
12	Release	18. Zu liefernde Ergebnisse ausliefern 19. Release Rückblick halten

Tabelle 1-1: Übersicht über Scrum Prozesse

Diese Phasen beschreiben jeden Prozess im Detail und schließen jeweils die mit ihnen verbundenen Inputs, Hilfsmittel und Outputs ein. In jedem Prozess sind einige Inputs, Hilfsmittel, und Outputs verbindlich (diese haben ein Sternchen [*] nach deren Namen), während andere optional sind. Ob die optionalen Inputs, Hilfsmittel, und/oder Outputs einbezogen werden, kommt auf das jeweilige Projekt, die Organisation oder die Industrie an. Inputs, Hilfsmittel, und Outputs, die als verbindlich gekennzeichnet sind, sind wichtig für die erfolgreiche Implementierung von Scrum in jeder Organisation.

Für große Scrum Projekte, die eine Koordinierung über mehrere Teams erfordern, werden zusätzliche Inputs, Hilfsmittel und Outputs benötigt, die in Kapitel 13 - Scrum für große Projekte skalieren - definiert sind.

Es gibt auch spezifische Prozesse, die die Anwendung von Scrum auf der Ebene des Unternehmens definieren. Diese sind in Kapitel 14 enthalten - Scrum für das Unternehmen skalieren.

1.4.4.1 Initiierung

1. *Projekt Vision erstellen* — In diesem Prozess wird die wirtschaftliche Begründung geprüft, um eine Projektvisionserklärung zu erstellen, die als Inspiration dient und den Fokus für das gesamte Projekt liefert. Der Product Owner wird in diesem Prozess identifiziert.
2. *Scrum Master und Business Stakeholdern identifizieren* — In diesem Prozess werden der Scrum Master und der/die Business Stakeholdern anhand spezieller Auswahlkriterien identifiziert.
3. *Scrum Team bilden* — In diesem Prozess werden die Scrum Teammitglieder identifiziert. Normalerweise hat der Product Owner die primäre Verantwortung für die Auswahl der Teammitglieder, aber er oder sie übernimmt dies oft in Zusammenarbeit mit dem Scrum Master.
4. *Epic(en) entwickeln* — In diesem Prozess dient die Projektvisionserklärung als Grundlage dazu, Epicen zu entwickeln. Anwendergruppen Meetings können abgehalten werden, um Epicen zu entwickeln.
5. *Priorisiertes Produkt Backlog erstellen* — In diesem Prozess werden Epic(en) verfeinert und ausgearbeitet, dann priorisiert, um ein Priorisiertes Produkt Backlog für das Projekt zu erstellen. Die Done-Kriterien werden ebenfalls zu diesem Zeitpunkt festgelegt.
6. *Release-Planung durchführen* — In diesem Prozess überprüft das Scrum Kernteam die im Priorisierten Produkt Backlog enthaltenen User Storys, um einen Release-Plan zu entwickeln. Dieser stellt im Wesentlichen einen schrittweisen Bereitstellungszeitplan dar, der den Business Stakeholdern zur Verfügung gestellt werden kann. In diesem Prozess wird auch die Länge der Sprints festgelegt.

1.4.4.2 Planung und Abschätzung

7. *User Storys Erstellen* — In diesem Prozess werden User Storys und die damit verbundenen User Story Abnahmekriterien erstellt. User Storys werden in der Regel vom Product Owner geschrieben und sollen sicherstellen, dass die Anforderungen des Kunden deutlich und für die Business Stakeholdern verständlich dargestellt worden sind. User Story Schreib-Workshops können abgehalten werden, in denen Scrum Teammitglieder die User Storys erstellen. User Storys werden in das priorisierte Produkt Backlog eingebracht.

8. *User Storys abschätzen* — Bei diesem Prozess erklärt der Product Owner User Storys damit der Scrum Master und das Scrum Team den Arbeitsaufwand abschätzen können, welcher benötigt wird, um die Funktionalität, die in jeder User Story beschrieben wird, zu entwickeln.
9. *User Storys zusagen* — Bei diesem Prozess sagt das Scrum Team zu, die vom Product Owner für einen Sprint vorgesehenen User Storys zu liefern. Das Ergebnis dieses Prozesses sind zugesagte User Storys.
10. *Aufgaben identifizieren* — In diesem Prozess werden die zugesagten User Storys auf spezifische Aufgaben heruntergebrochen und in Form einer Aufgabenliste zusammengestellt.
11. *Aufgaben abschätzen* — In diesem Prozess schätzt das Scrum Kernteam, den benötigten Arbeitsaufwand ab, um jede Aufgabe der Aufgabenliste umzusetzen. Das Ergebnis dieses Prozesses ist eine aufwands-abgeschätzte Aufgabenliste.
12. *Sprint Backlog aktualisieren* — In diesem Prozess hält das Scrum Kernteam Sprint Planungsmeetings, in denen die Gruppe ein Sprint Backlog erstellt, das alle Aufgaben enthält, die im Sprint abgeschlossen werden sollen

1.4.4.3 Implementierung

13. *Zu liefernde Ergebnisse erzeugen* — In diesem Prozess arbeitet das Scrum Team an den Aufgaben im Sprint Backlog, um die zu liefernden Ergebnisse des Sprints zu erzeugen. Dabei wird oft ein Scrumboard benutzt, um die ausgeführten Arbeiten und Aktivitäten festzuhalten. Angelegenheiten oder Probleme, die dem Scrum Team begegnen, können in einem Impediment Logbuch festgehalten werden.
14. *Tägliche Standups durchführen* — In diesem Prozess wird jeden Tag ein hochkonzentriertes Meeting mit einem harten zeitlichen Anschlag durchgeführt. Man spricht von diesem Meeting als tägliches Standup Meeting. Es ist ein Forum für das Scrum Team, bei dem sich die Teammitglieder gegenseitig über ihren Fortschritt und jegliche Hindernisse informieren, denen sie gegebenenfalls begegnen.
15. *Priorisiertes Produkt Backlog verfeinern* — In diesem Prozess wird das priorisierte Produkt Backlog ständig aktualisiert und gepflegt. Ein priorisiertes Produkt Backlog Meeting kann abgehalten werden. In diesem Meeting werden jegliche Änderungen oder Aktualisierungen des Backlogs besprochen und wenn angemessen in das priorisierte Produkt Backlog mit eingebunden.

1.4.4.4 Review und Rückblick

16. *Sprint vorführen und validieren* — In diesem Prozess führt das Scrum Team die zu liefernden Ergebnisse des Sprints dem Product Owner und den relevanten Business Stakeholdern in einem Sprint Rückblick Meeting vor. Der Zweck dieses Meetings ist es, sich vom Product Owner die Zustimmung und Akzeptanz für die zu liefernden Ergebnisse, die in dem Sprint hergestellt wurden, geben zu lassen.
17. *Sprint Rückblick halten* — In diesem Prozess treffen sich Scrum Master und Scrum Team, um über die Lehren zu diskutieren, die im Verlauf des Sprints gezogen wurden. Diese werden dokumentiert und können bei zukünftigen Sprints zur Anwendung kommen. Vereinbarte Verbesserungsmaßnahmen oder eine Aktualisierung der Scrum Guidance Body Empfehlungen sind des Öfteren Resultate dieser Besprechungen.

1.4.4.5 Release

18. *Zu liefernde Ergebnisse ausliefern* — In diesem Prozess werden akzeptierte zu liefernde Ergebnisse ausgeliefert oder an den zuständigen Business Stakeholdern weitergeleitet. Eine formale Vereinbarung über funktionierende zu liefernde Ergebnisse dokumentiert den erfolgreichen Abschluss des Sprints.
19. *Release Rückblick halten* — In diesem Prozess, der das Projekt abschließt, kommen organisatorische Business Stakeholdern und die Mitglieder des Scrum Kernteams zusammen, um auf das Projekt zurückzublicken und Lehren aus dem Projekt zu identifizieren, zu dokumentieren und zu verinnerlichen. Oft führen diese Lehren zur Dokumentation von vereinbarten Verbesserungsmaßnahmen, die in zukünftigen Projekten umgesetzt werden.

1.4.4.6 Scrum Meetings oder Zeremonien

Scrum Meetings oder Zeremonien spielen eine entscheidende Rolle bei der effektiven Implementierung von Scrum und sind ein zentrales Mittel, um die Scrum Prinzipien zu implementieren. Die wichtigen Scrum Meetings und die damit verbundenen Prozesse, in denen diese Meetings durchgeführt werden, sind in Tabelle 1-2 zusammengefasst.

Scrum Meetings	Scrum Prozesse
Projektvisions-Meeting	<ul style="list-style-type: none"> • Projektvision erstellen
Anwender-Gruppen Meetings	<ul style="list-style-type: none"> • Epics entwickeln • User Storys erstellen
Fokusgruppen Meetings	<ul style="list-style-type: none"> • Epics entwickeln • User Storys erstellen
Release Planungs-Sessions oder Meetings	<ul style="list-style-type: none"> • Release Planung durchführen
Produkt Backlog Review Meetings	<ul style="list-style-type: none"> • User Storys abschätzen • Priorisiertes Produkt Backlog verfeinern
Sprint Planungs-Meetings	<ul style="list-style-type: none"> • User Storys abschätzen • User Storys zusagen • Aufgaben identifizieren • Aufgaben abschätzen • Sprint Backlog aktualisieren
Tägliches Standup Meeting	<ul style="list-style-type: none"> • Tägliches Standup durchführen
Sprint Review Meeting	<ul style="list-style-type: none"> • Sprint durchführen und validieren
Sprint Retrospektions-Meeting	<ul style="list-style-type: none"> • Sprint Rückblick halten
Release Rückblick Meeting	<ul style="list-style-type: none"> • Release Rückblick halten

Tabelle 1-1: Scrum Meetings und Prozesse

1.4.4.7 Scrum für große Projekte Skalieren

Bei großen Projekten, die den Einsatz mehrerer (vier oder mehr) Scrum-Teams mit mehreren Product Ownern und mehreren Scrum-Mastern erfordern, bleiben die in den Kapiteln 8 bis 12 definierten grundlegenden Prozesse gültig, aber es können zusätzliche Überlegungen und Aktualisierungen der Inputs, Tools und Outputs erforderlich sein. Dies kann zusätzliche Koordinations- und Synchronisationserfordernisse beinhalten. Die Auswirkungen auf die grundlegenden Scrum-Prozesse bei der Skalierung von Scrum auf große Projekte werden in Kapitel 13 beschrieben.

Die Definition, was ein großes Projekt ist, hängt in der Regel von der Organisation und/oder der Komplexität der durchgeführten Projekte ab. Ein Schlüsselkriterium dafür, ob ein Projekt als klein oder groß gilt, ist, ob das Projekt mehrere Scrum Master und/oder mehrere Product Owner erfordert. Wenn das Projekt nur einen Scrum Master und einen Product Owner erfordert, dann können diese Personen normalerweise alle zusätzlichen Kommunikations- und Synchronisationsanstrengungen bewältigen, die das Projekt erfordert.

1.4.4.8 Scrum für das Unternehmen Skalieren

Bei der Anwendung von Scrum auf Unternehmensebene (z.B. auf ein Programm oder Portfolio) kann es mehrere hundert Scrum-Teams geben, in denen mehrere tausend Personen an mehreren Projekten innerhalb von Programmen und/oder Portfolios in der gesamten Organisation arbeiten. Die Anwendung von Scrum auf Programm- oder Portfolioebene hat bestimmte Auswirkungen auf die zugrunde liegenden Projekte. Im Allgemeinen sollten die Scrum-Projekte immer noch unter Verwendung der grundlegenden Scrum-Prozesse durchgeführt werden, die in den Kapiteln 8 bis 12 für typische kleine Projekte besprochen wurden, während die zusätzlichen Überlegungen, die in Kapitel 13 für große Projekte (mit mehreren Product Ownern und/oder Scrum Mastern) beschrieben werden, einbezogen werden.

Einige der Herausforderungen, die auf Programm- oder Portfolioebene auftreten, ähneln denen, die sich bei der Anwendung von Scrum auf ein großes Projekt ergeben. Die Synchronisierung zwischen den Teams und die allgemeine Zusammenarbeit sind in der Regel die größten Herausforderungen bei einem großen Scrum-Projekt - dies ist auch eine Herausforderung bei der Anwendung von Scrum auf Programm- oder Portfolioebene. Die größten Herausforderungen bei der Anwendung von Scrum auf Programm- oder Portfolioebene treten jedoch in der Regel auf der geschäftlichen Seite auf, weil die geschäftlichen Prioritäten der verschiedenen Projekte sowohl untereinander als auch mit den Gesamtzielen des Programms oder Portfolios in Konflikt geraten können. Diese Prioritäten und Ziele müssen in Einklang gebracht werden.

Bei der Implementierung von Scrum auf Unternehmensebene gibt es nicht nur zusätzliche Inputs, Tools und Outputs wie bei einem großen Scrum-Projekt, sondern auch spezifische zusätzliche Prozesse, die erforderlich sind, um die zusätzlichen Priorisierungs-, Abgleich- und Koordinierungsbemühungen zu bewältigen. Diese zusätzlichen Überlegungen werden in Kapitel 14 erörtert.

1.5 Scrum versus traditionelles Projektmanagement

Tabelle 1-3 fasst viele Unterschiede zwischen Scrum und den traditionellen Projektmanagement Modellen zusammen.

	Scrum	Traditionelles Projekt Management
Schwerpunkt auf	Personen	Prozessen
Dokumentation	Minimal — nur soweit erforderlich	Umfassend
Prozessstil	Iterativ	Linear
Planung im Voraus	Niedrig	Hoch
Priorisierung auf Anforderungen	Auf dem geschäftlichen Wert basierend und regelmäßig aktualisiert	Im Projektplan festgesetzt
Qualität und Qualitätssicherung	Der Kunde steht im Mittelpunkt	Der Prozess steht im Mittelpunkt
Organisation	Selbstorganisierend	Gemanagt
Managementstil	Dezentralisiert	Zentralisiert
Änderung	Aktualisierung des priorisierten Produkt Backlog	Formelles Änderungs Management System
Führung und Leitung	Gemeinschaftlich, dienende Führung	Befehl und Kontrolle
Erfolgsmessung	Geschäftlicher Wert	Planungs Konformität
Investitionsrendite	Früh / durchgängig im Projektzeitraum	Am Ende des Projekts
Kundenbeteiligung	Hoch durchgängig während des gesamten Projektzeitraums	Unterschiedlich je nach Projektzyklus

Tabelle 1-2: Scrum versus traditionelles Projektmanagement

2. PRINZIPIEN

2.1 Einführung

Die Scrum-Prinzipien sind die Grundlage des Scrum-Rahmens. Die Prinzipien von Scrum können auf jede Form von Projekt oder Organisation angewendet werden, und sie müssen beachtet werden, um die korrekte Verwendung von Scrum zu sichern. Während die Aspekte und Prozesse von Scrum modifiziert werden können um die Anforderungen des Projektes zu erfüllen, sind die Scrum-Prinzipien unverrückbar und müssen genau auf die Weise angewendet werden, wie sie im *Scrum Body of Knowledge (SBOK®)* beschrieben sind. Die Aufrechterhaltung der Prinzipien und ihre korrekte Anwendung sorgt beim Nutzer des Scrum-Rahmens für Vertrauen im Bezug auf das Erreichen der Ziele des Projekts. Prinzipien werden als die Schlüsselrichtlinien für die Anwendung des Scrum-Konzepts angesehen.

Prinzipien können nach der Definition des *Scrum Body of Knowledge (SBOK®)* auf folgendes angewendet werden:

- Portfolios, Programme und/oder Projekte in *jeder* Branche
- Produkte, Dienstleistungen, oder alle anderen Ergebnisse, welche an die Business Stakeholdernn geliefert werden soll.
- Projekte jeder Größe und Komplexität

Der Begriff "Produkt" in dieser Wissenssammlung kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann auf jedes Projekt in jeder Branche effektiv angewendet werden — von kleinen Projekten oder Teams mit lediglich sechs Team-Mitgliedern bis hin zu großen, komplexen Projekten mit mehreren Hundert Team-Mitgliedern.

Dieses Kapitel unterteilt sich in die folgenden Abschnitte:

2.2 Leitfaden für Kernrollen — Dieser Abschnitt skizziert welche Abschnitte oder Unterabschnitte besonders relevant für die Scrum Kernrollen Product Owner, Scrum Master und Scrum Team sind.

2.3 Empirische Prozesssteuerung — Dieser Abschnitt beschreibt das erste Prinzip von Scrum, und die drei Hauptideen von Transparenz, Inspektion und Adaptierung.

2.4 Selbstorganisation — Dieses Prinzip ist auf den modernen Angestellten fokussiert, welcher wesentlich bessere Ergebnisse liefert, wenn er selbstorganisiert agieren kann, was zu einer größeren Teamakzeptanz und gemeinsamer Verantwortung führt; darüber hinaus resultiert es auch in einer innovativen und kreativen Umgebung welche förderlicher für Wachstum ist.

2.5 Kollaboration — Dieser Abschnitt betont das dritte Prinzip von Scrum, wonach die Produktentwicklung ein gemeinsamer Wertzeugender Prozess ist, bei welchem die Zusammenarbeit aller Business Stakeholdern zur Lieferung des größten Werts notwendig ist. Der Abschnitt fokussiert auch auf die Kern-Elemente der Zusammenarbeit: Bewusstsein, Zergliederung und Aneignung.

2.6 Wertbasierte Priorisierung — Dieser Abschnitt präsentiert das vierte Prinzip von Scrum, welches den Antrieb des Scrum-Rahmens zur Lieferung des maximalen Geschäftswerts in der kürzestmöglichen Zeit beschreibt.

2.7 Time-Boxing — Dieser Abschnitt erklärt das fünfte Prinzip von Scrum, welches Zeit als die limitierende Beschränkung behandelt. Er deckt dabei den Sprint, das tägliche Standup Meeting und verschiedene andere Sprintbezogene Meetings wie das Sprint-Planungs Meeting, das Sprint-Review Meeting und das Sprint Retrospektions-Meeting ab, auf die alle Time-Boxing angewendet wird.

2.8 Iterative Entwicklung — Dieser Abschnitt behandelt das sechste Prinzip von Scrum, welches die iterative Entwicklung hervorhebt, welche dabei hilft, besser mit Änderungen umzugehen und das Produkt zu entwickeln, das der Kunde wirklich benötigt.

2.9 Scrum versus traditionelles Projektmanagement — Dieser Abschnitt betont die Hauptunterschiede zwischen den Scrum-Prinzipien und traditionellen Projektmanagement-Prinzipien (Wasserfallmodell) und erklärt, warum Scrum in der heutigen, sich schnell verändernden Zeit besser funktioniert.

2.2 Leitfaden für Kernrollen

Alle Abschnitte in diesem Kapitel sind für alle Rollen im Scrum Kernteam wichtig - für den Product Owner, den Scrum Master, und das Scrum Team. Ein klares Verständnis der Scrum-Prinzipien durch alle Business Stakeholdern ist für den Erfolg des Scrum-Konzepts in jeder Organisation von entscheidender Bedeutung.

2.3 Empirische Prozesssteuerung

Bei Scrum werden Entscheidungen anstelle von umfangreichen vorherigen Planungen auf Grund von Beobachtung und Experimentieren getroffen. Empirische Prozesssteuerung stützt sich auf die drei Grundideen der Transparenz, Inspektion und Adaptierung.

2.3.1 Transparenz

Transparenz ermöglicht das Beobachten von allen Facetten eines jeden Scrum-Prozesses durch jeden Beteiligten. Dies fördert einen einfachen und durchsichtigen Fluss von Informationen innerhalb der gesamten Organisation und führt zu einer offenen Arbeitsatmosphäre. Transparenz wird durch folgende Aspekte wiedergegeben:

- Eine Projektvisionserklärung, welche von allen Business Stakeholdern und dem Scrum Team eingesehen werden kann
- Ein offenes priorisiertes Produkt Backlog mit priorisierten User Storys, welches von jedem, ob innerhalb oder außerhalb des Scrum Teams, eingesehen werden kann

- Ein Release Plan, welcher zur Koordinierung über mehrere Scrum Teams hinweg und mit den Business Stakeholdern verwendet werden kann
- Klare Einsicht in die Fortschritte des Teams anhand der Verwendung eines Scrumboards, eines Burndown-Charts, und anderer Hilfsmittel zur Informationsverbreitung
- Sprint Planungs-Meetings, in denen das Scrum Team den Aufwand für die Entwicklung von User Storys von hoher Priorität abschätzt und die Fertigstellung einer Reihe von User Storys im Sprint zusagt.
- *Tägliche Standup Meetings*, die im Prozess *Tägliches Standup durchführen* stattfinden, und in denen alle Teammitglieder berichten, woran sie am Vortag gearbeitet haben, was sie für heute planen, und ob irgendwelche Probleme bestehen, welche sie von der Fertigstellung ihrer Aufgaben im gegenwärtigen Sprint abhalten.
- Sprint Review Meetings, welche während Sprint vorführen und validieren durchgeführt werden, in denen das Scrum Team dem Product Owner und den Business Stakeholdern potenziell versandfähige zu liefernde Ergebnisse vorführt.
- Sprint Retrospektions-Meetings, die nach dem Sprint Review am letzten Tag des Sprints durchgeführt werden, und in denen das Scrum Team mögliche Verbesserungen für zukünftige Sprints bespricht.

Abbildung 2-1 fasst das Konzept der Transparenz bei Scrum zusammen.

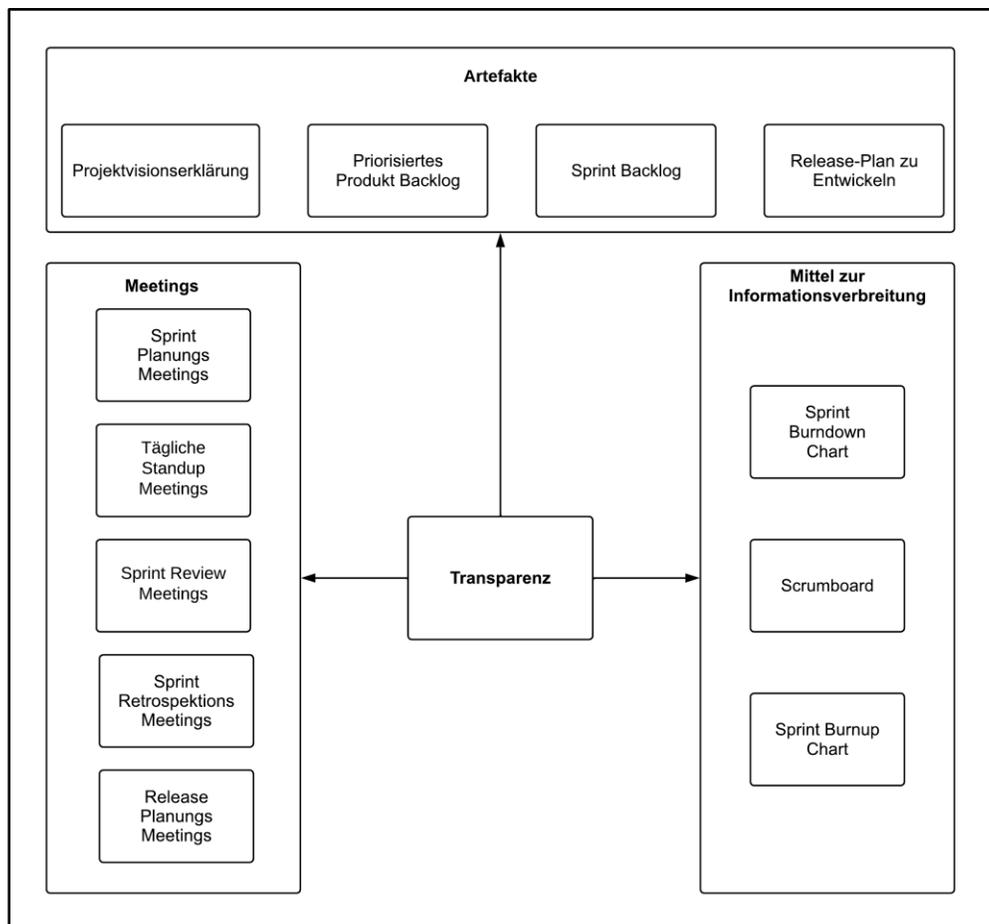


Abbildung 2-1: Transparenz bei Scrum

2.3.2 Inspektion

Inspektion bei Scrum wird durch folgende Aspekte wiedergegeben:

- Verwendung eines Scrumboards oder eines anderen Hilfsmittels zur Informationsverbreitung, welche den Fortschritt des Scrum Teams bei der Fertigstellung der Aufgaben im gegenwärtigen Sprint wiedergeben.
- Sammlung von Feedback von den Kunden und anderen Business Stakeholdern während der Prozesse *Epics entwickeln*, *Priorisiertes Produkt Backlog erstellen*, und *Release-Planung durchführen*.
- Inspektion und Genehmigung der zu liefernden Ergebnisse durch den Product Owner und die Kunden bei *Sprint vorführen und validieren*.

Abbildung 2-2 fasst das Konzept der Inspektion bei Scrum zusammen.

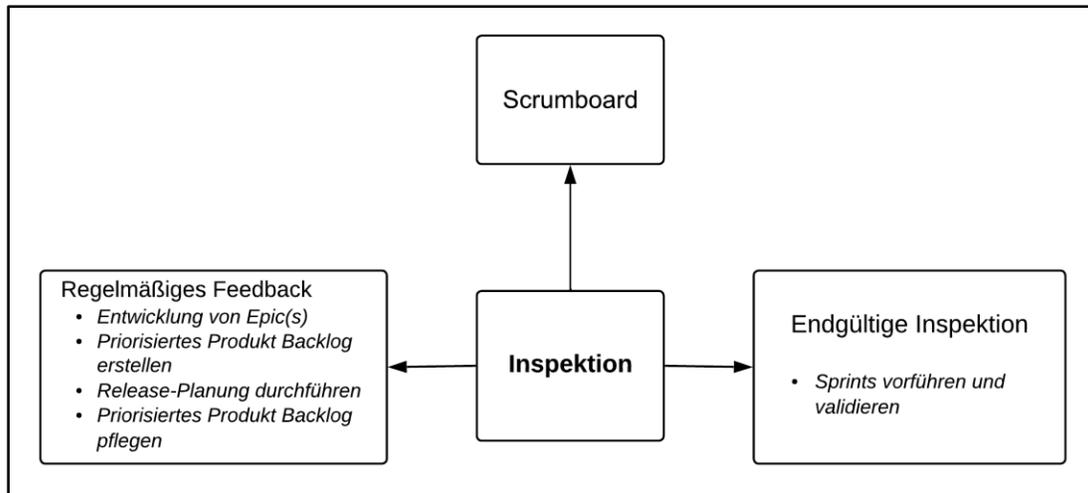


Abbildung 2-2: Inspektion bei Scrum

2.3.3 Adaptierung

Adaptierung geschieht, wenn das Scrum Kernteam und die Business Stakeholdern durch Transparenz und Inspektion lernen und dann ihre Arbeitsweise an die erlangten Erkenntnisse anpassen. Einige Beispiele für Möglichkeiten zur Adaptierung in Scrum sind:

- In täglichen Standup Meetings diskutieren Scrum Team-Mitglieder offen Hindernisse für die Erfüllung ihrer Aufgaben und suchen nach Hilfe bei den anderen Teammitgliedern. Erfahrenere Mitglieder des Scrum Teams leiten darüber hinaus die mit dem Projekt oder der Technik weniger erfahrenen Teammitglieder an.

- Risikoidentifikation wird während des gesamten Projekts durchgeführt und wiederholt. Identifizierte Risiken werden zu Inputs bei mehreren Scrum-Prozessen, darunter *Priorisiertes Produkt Backlog erstellen*, *Priorisiertes Produkt Backlog verfeinern*, und *Sprint durchführen und validieren*.
- Verbesserungen können auch zu Change Requests führen, welche während *Epic(s) entwickeln*, *Priorisiertes Produkt Backlog erstellen*, und *Priorisiertes Produkt Backlog verfeinern* besprochen und genehmigt werden.
- Der Scrum Guidance Body interagiert mit dem Scrum Team während *User Storys erstellen*, *Aufgaben abschätzen*, *Zu liefernde Ergebnisse erzeugen* und *Priorisiertes Produkt Backlog verfeinern*, um Anleitung und Expertise zu bieten, so diese benötigt wird.
- Bei *Sprint-Rückblick durchführen* werden vereinbarte Verbesserungsmaßnahmen aufgrund der Ergebnisse aus *Sprint durchführen und validieren* festgelegt.
- Beim Releaserückblick-Meeting dokumentieren die Teilnehmer die gelernten Lektionen und führen Reviews durch, um Gelegenheiten zur Verbesserung und Korrektur von Unwirtschaftlichkeiten zu finden.

Abbildung 2-3 fasst das Konzept der Adaptierung bei Scrum zusammen.

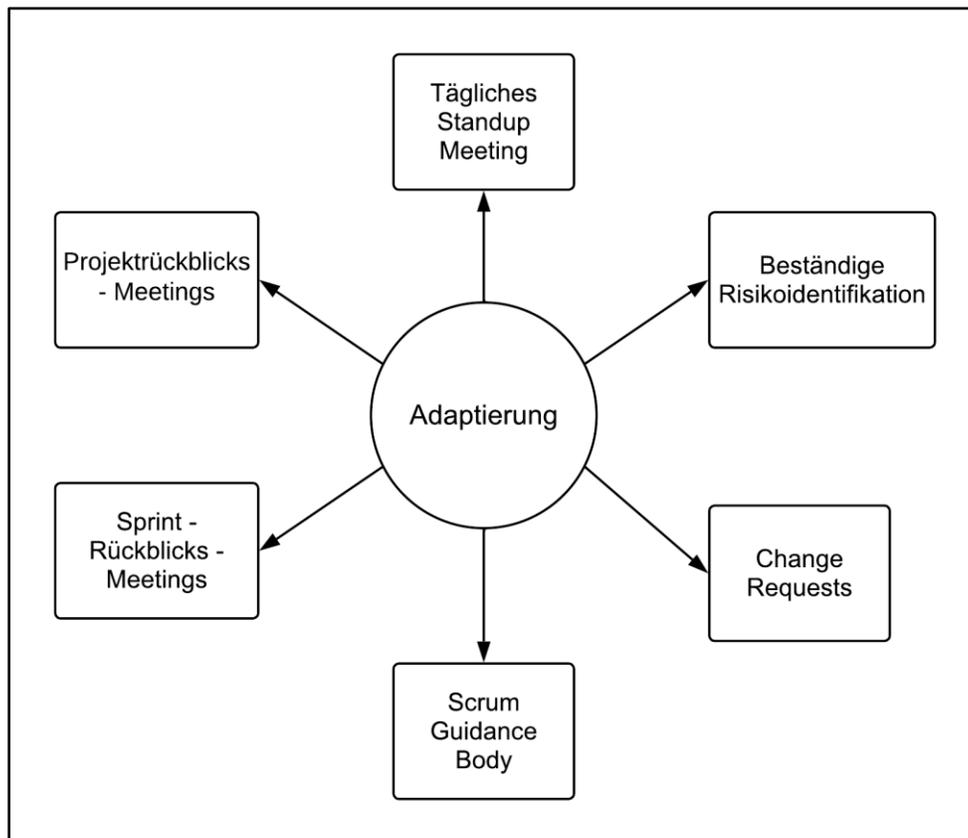


Abbildung 2-3: Adaptierung in Scrum

Bei anderen Methoden, wie beispielsweise dem traditionellen Wasserfallmodell, ist ein beachtliches Maß an vorheriger Planung erforderlich, und der Kunde erhält üblicherweise nicht vor Ende einer Phase, oder gar vor Ende des gesamten Projekts, die Möglichkeit die Komponenten des Produkts zu überprüfen. Diese Methode

beinhaltet oft große Risiken für den Erfolg des Projekts, da sie ein größeres Potential hat, die Lieferung des Projekts und die Akzeptanz bei den Kunden zu beeinflussen. Die Interpretation und das Verständnis des Kunden vom fertigen Produkt kann deutlich von demjenigen abweichen, was vom Team verstanden und produziert wird, ein Fakt, der mitunter nicht vor einer sehr späten Phase in der Entwicklung des Produkts bekannt wird.

Abbildung 2-4 demonstriert ein Beispiel für solche Probleme.

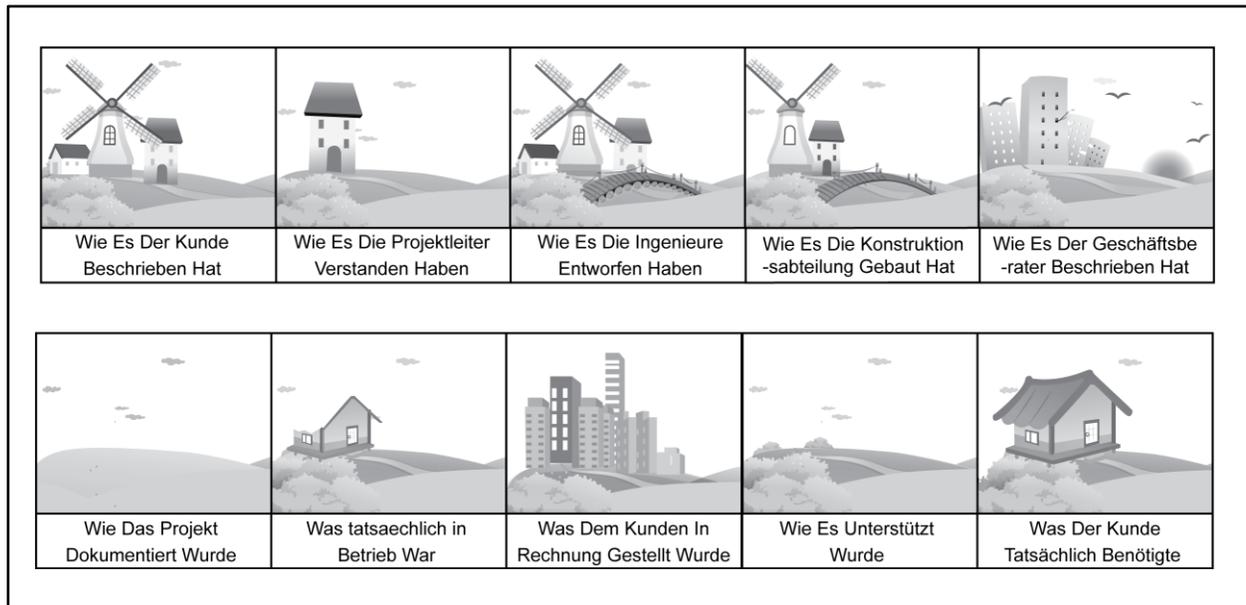


Abbildung 2-4: Probleme beim Wasserfall-Projektmanagement

2.4 Selbstorganisation

Scrum glaubt daran, dass Beschäftigte selbstmotiviert sind und nach größerer Verantwortung streben. Daher werden sie größere Werte abliefern, wenn sie sich selbst organisieren können.

Der bevorzugte Führungsstil bei Scrum ist die "Unterstützende Führung", welcher das Erreichen von Ergebnissen betont, in dem er sich auf die Bedürfnisse des Scrum Teams konzentriert. Für eine Betrachtung der verschiedenen Führungs- und Managementstile siehe auch Abschnitt 3.10.3.

Selbstorganisation bedeutet nicht, dass die Teammitglieder sich so verhalten können, wie es ihnen gerade in den Sinn kommt. Sobald im Prozess *Produktvision erstellen* die Projektvision erstellt wurde, werden der Product Owner, Scrum Master und das Scrum Team identifiziert. Das Scrum Core Team selbst arbeitet sehr eng mit den relevanten Business Stakeholdern zur besseren Verfeinerung der Anforderungen während *Epic(s) entwickeln* und *User Storys erstellen* zusammen. Fachkenntnisse des Teams werden dazu genutzt, die Vorgaben zur Ausführung der geplanten Arbeit am Projekt zu bewerten. Diese Bewertung und Expertise wird auf alle technischen Aspekte und auf Aspekte des Managements während *zu liefernden Ergebnisse erzeugen* angewandt.

Obwohl die Priorisierung primär durch den Product Owner, welcher die Stimme des Kunden repräsentiert, geschieht, ist das selbstorganisierte Scrum Team am Aufteilen und Abschätzen der *Aufgaben bei deren Erstellung* und *Abschätzung* beteiligt. Dabei ist jedes Teammitglied dafür verantwortlich, zu bestimmen, welche Arbeit sie/er übernimmt. Das Scrum Team unterstützt den Product Owner bei der Identifikation von Risiken und Abhängigkeiten. Bei der Durchführung des Sprints wird, wenn die Teammitglieder Hilfe benötigen, dies bei der verbindlichen regelmäßigen Interaktion beim täglichen Standup Meeting behandelt. Das Scrum Team selbst interagiert mit anderen Teams anhand des Scrum of Scrums Meetings und kann für weitere Anleitung, so nötig, den Scrum Guidance Body zu Rate ziehen.

Schließlich arbeiten Scrum Team und Scrum Master während Sprint vorführen und validieren, bei dem korrekt verfertigte zu liefernde Ergebnisse akzeptiert werden, eng zusammen, um den Teil des Projektes, welcher im Sprint erzeugt wurde, vorzustellen. Da zu liefernde Ergebnisse potenziell versandfähig sind (und da die User Storys im priorisierten Produkt Backlog in der Reihenfolge der durch sie erzeugten Werte priorisiert sind), können Product Owner und der Kunde den Wert, der bei jedem Sprint generiert wird, klar visualisieren und ausdrücken; das Scrum Team wiederum erhält die Genugtuung zu sehen, dass seine harte Arbeit vom Kunden und anderen Business Stakeholdern akzeptiert wird.

2.4.1 Benefits der Selbstorganisation

Selbstorganisation als ein essentielles Prinzip von Scrum führt zu:

- Zustimmung des Teams und gemeinsame Verantwortung
- Motivation, welche zu einem verbesserten Leistungsstand des Teams führt.
- Innovatives und kreatives Arbeitsumfeld, das wachstumsfördernd ist.

Die Hauptziele von selbstorganisierenden Teams lauten wie folgt:

- Auswahl des einfachsten und besten Ansatzes, mit dem die Anforderungen erfüllt werden können
- Verstehen der Projektvision und weshalb das Projekt für die Organisation von Wert ist
- Abschätzung der User Storys in *User Storys abschätzen* und Zuordnung von Aufgaben an sich selbst bei *Sprint Backlog aktualisieren*.
- Selbständiges Erstellen von Aufgaben während *Aufgaben identifizieren*.
- Anwendung und Nutzbarmachung der eigenen Expertise als ein funktions-übergreifendes Team, um an den Aufgaben während *Zu liefernde Ergebnisse erzeugen* zu arbeiten.
- Lieferung von greifbaren Ergebnissen, welche von Kunden und Business Stakeholdern bei *Sprint vorführen und validieren* akzeptiert werden.
- Lösung individueller Probleme zusammen mit der Behandlung derselben während der täglichen Standup Meetings.
- Klärung jeglicher Diskrepanzen oder Zweifel und Offenheit für das Erlernen von Neuem.
- Kontinuierliche Aktualisierung von Wissen und Fähigkeiten mittels Interaktion innerhalb des Teams.
- Erhaltung der Stabilität der Teammitglieder während der gesamten Dauer des Projekts, in dem keine Mitglieder ausgetauscht werden, wenn dies nicht zwingend erforderlich ist.

Abbildung 2-5 stellt die Ziele eines sich selbst organisierenden Teams dar.

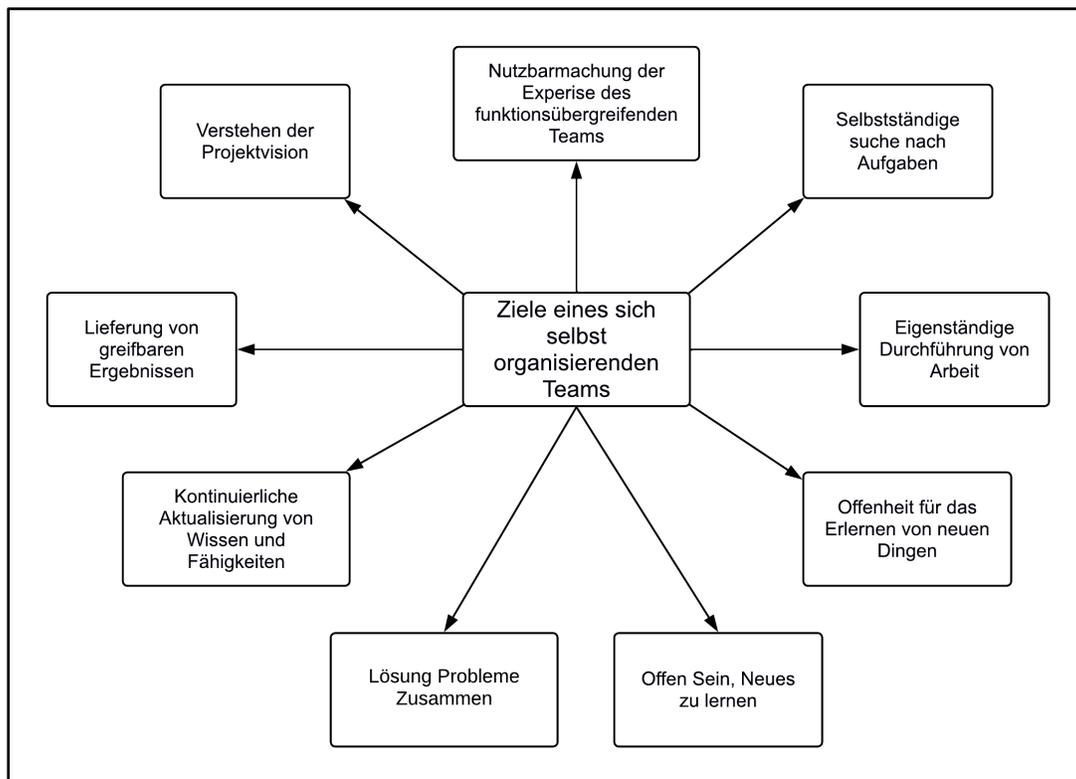


Abbildung 2-5: Ziele eines sich selbst organisierenden Teams

2.5 Kollaboration

Kollaboration bezieht sich im Rahmen vom Scrum auf die Zusammenarbeit des Scrum Kernteams und dessen Absprache mit den Business Stakeholdern, um zu liefernde Ergebnisse des Projekts zu erzeugen und validieren, welche die in der Projektvision umrissenen Ziele erreichen. Es ist an dieser Stelle wichtig, den Unterschied zwischen Kooperation und Kollaboration herauszuarbeiten. Kooperation geschieht, wenn das Arbeitsprodukt aus der Summe des Arbeitsaufwands von verschiedenen Personen im Team besteht. Kollaboration geschieht, wenn ein Team zusammenarbeitet, um mittels der gegenseitigen Beiträge etwas Besseres zu entwickeln. Um eine umfassende Zusammenarbeit zu erreichen, ist es wichtig, Vertrauen zwischen allen Teammitgliedern sowie zwischen dem Team und den Business Stakeholdern aufzubauen.

Die Kernfaktoren kollaborativer Arbeit sind:

- *Bewusstsein* — Individuen, die zusammenarbeiten, müssen von der Arbeit des anderen wissen.
- *Zergliederung* — Kollaborierende Individuen müssen die Arbeit in Einheiten aufteilen, sie unter den Teammitgliedern verteilen, und sie nach erfolgter Arbeit wieder zusammenfügen.
- *Aneignung* — Adaptieren von Technologie auf die eigene Situation; die Technologie mag anders verwendet werden, als von den Entwicklern erwartet.

2.5.1 Vorteile der Kollaboration bei Scrum Projekten

Das Agile Manifest (Fowler & Highsmith, 2001) betont "Kundenkollaboration über Vertragsverhandlung." Dementsprechend adoptiert der Scrum-Rahmen einen Ansatz, bei dem die Scrum Core Team-Mitglieder (Product Owner, Scrum Master und Scrum Team) miteinander und mit den Business Stakeholdern zusammenarbeiten, um zu liefernde Ergebnisse mit dem größtmöglichen Wert für den Kunden zu erzeugen. Diese Kollaboration erfolgt während des gesamten Projekts.

Kollaboration sichert, dass die folgenden Vorteile für das Projekt umgesetzt werden:

1. Die Notwendigkeit von Änderungen aufgrund von schlecht geklärten Anforderungen wird minimiert. Während *Projektvision erstellen*, *Epic(s) entwickeln* und *Priorisiertes Produkt Backlog erstellen*, beispielsweise, kollaboriert der Product Owner mit den Business Stakeholdern um die Projektvision zu erstellen, Epics zu entwickeln und das Priorisierte Produkt Backlog zu erstellen. Dies führt zur Sicherstellung von Klarheit zwischen den Mitgliedern des Scrum Kernteams hinsichtlich der Arbeit, welche für die Fertigstellung des Projekts benötigt wird. Das Scrum Team kollaboriert fortwährend mit dem Product Owner und den Business Stakeholdern mittels eines transparenten priorisierten Produkt Backlogs, um die zu liefernden Ergebnisse des Projekts zu erstellen. In den Prozessen *Tägliches Standup Meeting durchführen*, *Priorisiertes Produkt Backlog verfeinern* und *Sprint-Rückblick durchführen* wird den Scrum Core Team-Mitgliedern Raum gegeben, das Getane zu besprechen und für dasjenige, das noch getan werden muss, zu kollaborieren. Auf diese Weise wird die Anzahl von Change Requests des Kunden und die Menge der Nacharbeit minimiert.

2. Risiken werden identifiziert und effizient gehandhabt. Risiken werden beispielsweise während *Epic(s) entwickeln, Zu liefernde Ergebnisse erzeugen und Tägliches Standup-Meeting durchführen* durch die Mitglieder des Scrum Kernteams identifiziert und bewertet. Die Scrum Hilfsmittel wie das tägliche Standup Meeting, das Sprint-Planungs Meeting, das Review Meeting des priorisierten Produkt Backlogs etc. geben dem Team Gelegenheiten, um Risiken nicht nur zu identifizieren und zu bewerten, sondern auch Reaktionen auf hochbrisante Risiken zu implementieren.
3. Das wahre Potential des Teams wird realisiert. Die Durchführung des täglichen Standup Meetings beispielsweise stellt Raum für das Scrum Team bereit, um zu kollaborieren und die Stärken und Schwächen seiner Mitglieder zu verstehen. Wenn ein Teammitglied eine Frist nicht einhalten kann, kollaboriert das Team miteinander, um die Aufgabe zu vollenden und die vereinbarten Ziele für die Fertigstellung des Sprints einzuhalten.
4. Kontinuierliche Verbesserung wird durch die gemachten Erfahrungen garantiert. Das Scrum Team nutzt beispielsweise die Durchführung des Sprint-Rückblicks, um zu klären, was beim vorausgehenden Sprint gut oder nicht gut gelaufen ist. Dies bietet eine Möglichkeit für den Scrum Master, mit dem Team daran zu arbeiten, das Team für den nächsten geplanten Sprint neu aufzusetzen und es zu verbessern. Dies sichert auch, dass Kollaboration im nächsten Sprint noch effektiver funktioniert.

Abbildung 2-6 verdeutlicht die Vorteile von Kollaboration bei einem Scrum-Projekt.

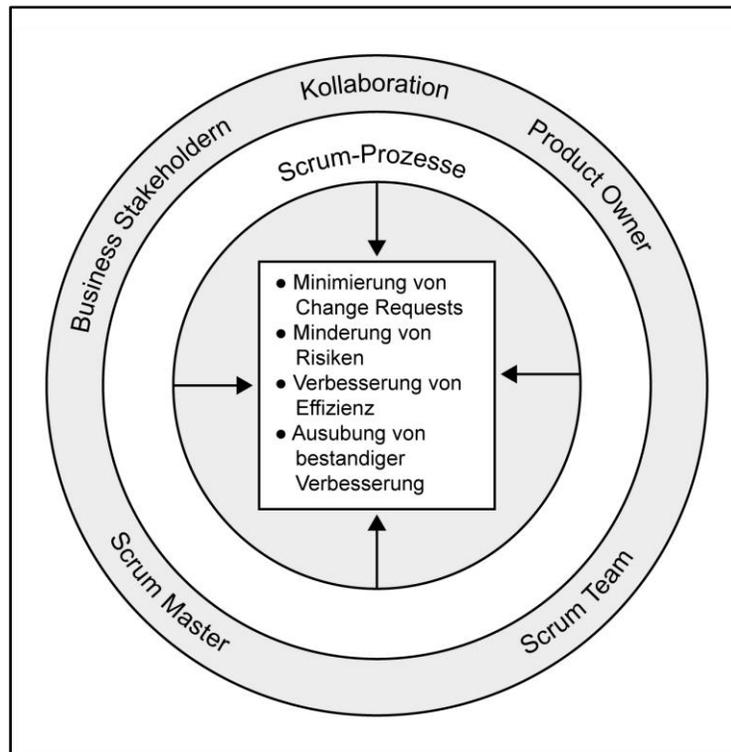


Abbildung 2-6: Vorteile der Kollaboration bei einem Scrum-Projekt

2.5.2 Die Bedeutung von Colocation für Kollaboration

Für viele Scrum-Praktiken ist intensive Kommunikation Grundvoraussetzung. Um diese zu ermöglichen, wird Colocation bevorzugt, d.h. dass die Teammitglieder an einem gemeinsamen physischen Ort sind, vorzugsweise im selben Büro. Das ermöglicht sowohl formelle als auch informelle Interaktion zwischen den Teammitgliedern. Dies sorgt für den Vorteil, die Teammitglieder immer für Koordination, Problemlösung, und Lernprozesse zur Hand zu haben. Einige der Vorteile von Colocation sind:

- Fragen werden schnell beantwortet
- Probleme können unmittelbar beseitigt werden
- Es gibt weniger Spannungen zwischen den Interaktionen
- Vertrauen wird schneller erworben und vergeben.

2.5.3 Zusammenarbeit in verteilten Teams

Obwohl kolokale Teams bevorzugt werden, kann es vorkommen, dass das Scrum-Team verteilt ist. Die Mitglieder des Scrum-Teams können an verschiedenen Orten arbeiten (z.B. in verschiedenen Städten oder sogar in verschiedenen Ländern) und/oder auch von zu Hause aus arbeiten. Selbst in einem kolokalen Team sollten die Teammitglieder die Flexibilität haben, aus der Ferne zu arbeiten, falls außergewöhnliche Umstände eintreten, die die Fähigkeit des Teams, in einer kolokalen Umgebung zu arbeiten, beeinträchtigen könnten. In solchen Situationen kann es erforderlich sein, sicherzustellen, dass die Mitglieder des Scrum-Teams so aufgestellt sind, dass sie effektiv mit dem verteilten Team arbeiten können.

2.5.3.1 Scrum-Projekttool

Die Verwendung eines kollaborativen Scrum Projekttools wird dringend empfohlen, um sicherzustellen, dass die Teammitglieder produktiv arbeiten können, insbesondere wenn die Teammitglieder nicht an einem gemeinsamen Arbeitsplatz sind. Das Tool sollte idealerweise die folgenden Möglichkeiten bieten

- Effiziente Definition aller Scrum-Rollen und Bereitstellung von Nachrichten-/Kollaborationsfunktionen für alle Teammitglieder zur Interaktion untereinander;
- Wichtige Scrum-Artefakte wie das Priorisierte Produkt-Backlog, das Sprint-Backlog, das Scrumboard usw. zu erstellen und zu bearbeiten;
- Bereitstellung des Workflows zur Bearbeitung verschiedener Scrum-Prozesse in den Phasen Initiierung, Planung und Abschaetzung, Implementierung, Review und Rückblick, und Release;
- Skalierung auf Organisations- oder Unternehmensebene (wenn Scrum in einer großen Organisation/einem großen Unternehmen implementiert wird);
- Planung von Scrum-bezogenen Meetings, wie Release Planungs-Meetings, Täglichen Standup Meetings, Sprint Planungs-Meetings, Sprint Review Meetings, Sprint Retrospektions-Meetings, Produkt Backlog Review Meetings, usw. Die eigentlichen Meetings können jedoch in einem separaten Videokonferenz-Tool durchgeführt werden;

- es den Mitgliedern des Scrum-Teams ermöglichen, auf einfache Weise online miteinander zu kommunizieren, entweder eins zu eins oder über verteilte Gruppen und/oder Diskussionsforen (da effektive Scrum-Teams mit gemeinsamen Standorten regelmäßig miteinander kommunizieren); (Beachten Sie jedoch, dass Teammitglieder in verteilten Teams im Gegensatz zu kolokalen Teams verstehen müssen, dass andere Teammitglieder möglicherweise nicht zur gleichen Zeit für eine sofortige Kommunikation verfügbar sind).
- Erfassung der gewonnenen Erkenntnisse (aus Retrospektiven usw.) mit entsprechenden Berichten, die im laufenden Betrieb erstellt werden;
- Automatisierung, damit alle Vorlagen oder Anleitungen des Scrum Guidance Body allen Scrum-Teams in der gesamten Organisation zur Verfügung stehen (z. B. die Definition von Ready oder die Definition von Done);
- dem SGB erlauben, Scrum-bezogenes Verhalten zu bewerten (z.B. maximale Anzahl von Teammitgliedern, Sprint-Dauer, usw.); und
- Klonen von ähnlichen Projekten, Epics und User Storys - dies ermöglicht es den Mitgliedern des Scrum Teams, weniger Zeit mit der Erstellung unnötiger oder doppelter Dokumentation zu verbringen und von den Erfahrungen aus ähnlichen bereits abgeschlossenen Arbeiten zu lernen. (Dies ist besonders vorteilhaft, wenn Scrum Teams ähnliche Implementierungsprozesse verwenden, um identische Produktkategorien zu erstellen, z.B. eine Werbefirma, die Anzeigenkampagnen für verschiedene Kunden erstellt; eine Baufirma, die Zeichnungen für ähnliche Straßenbauaktivitäten erstellt, usw.).

Zu den Vorteilen der Verwendung eines Scrum Projekttools für verteilte Teams gehören die folgenden

- Unterstützung der Arbeit von Scrum-Team-Mitgliedern wann sie wollen und von wo sie wollen
- Automatisierung von Berichten, Chats, Kalender, Workflows usw.
- Durchsetzung von Standardrichtlinien in der gesamten Organisation durch Automatisierung von SGB-Empfehlungen
- Erhöhte Effizienz durch Verringerung des Zeitaufwands für die Erstellung sich wiederholender oder unnötiger Dokumentation beim Klonen von ähnlichen Projekten (z. B. Klonen von Epics und User Storys)
- Die Arbeit mit einem vielfältigeren Team (manchmal aus verschiedenen Ländern) bringt oft lokale Perspektiven und Erfahrungen ein
- Weniger logistische Herausforderungen im Vergleich zu den Herausforderungen, die entstehen können, wenn alle Teammitglieder von einem Standort aus arbeiten. Dies kann auch Zeit und Kosten für Reisen, teure Arbeitsorte usw. sparen.

Für verteilte Teams ist es wichtig, den Scrum Prinzipien besondere Aufmerksamkeit zu schenken, um sicherzustellen, dass sie befolgt werden. Der Schwerpunkt sollte auf der Ermöglichung einer transparenten und kollaborativen Arbeitsumgebung des Vertrauens liegen.

2.6 Wertbasierte Priorisierung

Der Scrum-Rahmen wird von dem Ziel angetrieben, ein Maximum an Geschäftswert in einer möglichst kurzen Zeitspanne zu liefern. Eines der effektivsten Mittel für die Lieferung des größtmöglichen Wertes in der kürzestmöglichen Zeit ist Priorisierung.

Priorisierung kann als das Festlegen der Reihenfolge und das Trennen der Aufgaben, welche jetzt erledigt werden müssen, von denjenigen, welche später erledigt werden müssen, definiert werden. Das Konzept der Priorisierung ist innerhalb des Projektmanagements nicht neu. Das traditionelle Wasserfallmodell des Projektmanagement suggeriert die Verwendung von mehreren Mitteln der Aufgabenpriorisierung. Aus der Sicht des Projektmanagers ist Priorisierung von besonderer Bedeutung, da bestimmte Aufgaben zuerst erledigt werden müssen, um die Entwicklung zu beschleunigen und die Projektziele zu erreichen. Diese traditionellen Methoden umfassen unter anderem das Festlegen von Fristen und das Delegieren von Aufgaben anhand einer Priorisierungs-Matrix.

Bei Scrum ist jedoch die Wertbasierte Priorisierung eines der Kernprinzipien, welche die Struktur und Funktionalität des gesamten Scrum-Rahmens ausmachen; darüberhinaus hilft sie Projekten durch Adaptierfähigkeit und iterative Entwicklung des Produkts zu profitieren. Noch wichtiger ist, dass Scrum das Ziel hat, dem Kunden ein wertvolles Produkt zu einem frühen Zeitpunkt und auf kontinuierlicher Basis zu liefern. Priorisierung erfolgt durch den Product Owner, wenn dieser die User Storys im priorisierten Produkt Backlog priorisiert. Das priorisierte Produkt Backlog enthält eine Liste aller Anforderungen für ein Ertrag bringendes Projekt.

Sobald der Product Owner die geschäftlichen Anforderungen vom Kunden erhalten hat, werden diese in Form von Epics und User Storys (einem spezifischen Format zum Erfassen von Anforderungen) niedergeschrieben. Der Product Owner arbeitet mit dem Kunden und anderen Business Stakeholdern zusammen, um zu verstehen, welche geschäftlichen Anforderungen den maximalen Geschäftswert bieten. Mitunter kann es sein, dass ein Kunde alle User Storys als höchste Priorität festlegen will. Selbst eine Liste von User Storys mit höchster Priorität muss innerhalb der Liste priorisiert werden. Der Product Owner muss verstehen, was der Kunde will und wertschätzt, um die Einträge im priorisierten Produkt Backlog (die User Storys) nach ihrer relativen Wichtigkeit anzuordnen. Diese Liste wird Priorisiertes Produkt Backlog genannt und sollte alle bekannten Anforderungen für das Projekt enthalten. |Zur Priorisierung eines Backlogs gehört die Betrachtung, wie kritisch jede einzelne User Story ist. Anforderungen von hohem Wert werden identifiziert und an die Spitze des priorisierten Produkt Backlogs gesetzt. Die Prozesse, bei denen das Prinzip der wertbasierten Priorisierung zur Anwendung kommt sind *Priorisiertes Produkt Backlog erstellen* und *Priorisiertes Produkt Backlog verfeinern*.

Gleichzeitig muss der Product Owner mit dem Scrum Team zusammenarbeiten, um die Risiken und Ungewissheiten des Projekts zu verstehen, da diese mit negativen Auswirkungen verbunden sein können. Diese sollten während der Priorisierung der User Storys bei einem wertbasierten Ansatz (vergleiche das Kapitel zu Risiken für mehr Informationen) beachtet werden. Das Scrum Team benachrichtigt darüber hinaus den Product Owner über jegliche Abhängigkeiten, welche aus der Implementierung entstehen. Diese Abhängigkeiten müssen während der Priorisierung beachtet werden. Priorisierung kann auf einer subjektiven Einschätzung vom Geschäftswert und der Profitabilität beruhen, oder anhand der Ergebnisse und Analysen des Markts mittels Kundeninterviews, Umfragen, Finanzmodellen und analytischen Methoden etc. erfolgen.

Der Product Owner muss die Vorgaben und Bedürfnisse der Projekt Business Stakeholdern umsetzen, um das priorisierte Produkt Backlog zu erstellen. Daher beachtet er bei der Priorisierung der User Storys im priorisierten Produkt Backlog die folgenden drei Faktoren (vergleiche Abbildung 2-7):

1. Wert
2. Risiko oder Ungewissheit
3. Abhängigkeiten

Auf diese Weise resultiert Priorisierung nicht nur in zu liefernden Ergebnissen, welche die Anforderungen des Kunden erfüllen, sondern zielt auch darauf ab, maximalen Geschäftswert in der kürzestmöglichen Zeit zu liefern.

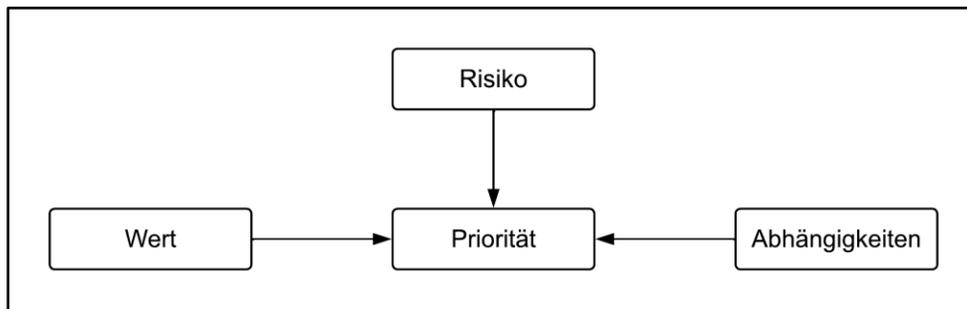


Abbildung 2-7: Wertbasierte Priorisierung

2.7 Time-boxing

Scrum behandelt Zeit als eine der wichtigsten Beschränkungen bei der Handhabung eines Projekts. Um die mit Zeit verbundenen Beschränkungen zu behandeln, verwendet Scrum ein Konzept namens 'Time-Boxing', welches die Festschreibung einer gewissen Zeitspanne für jeden Prozess und jede Aktivität in einem Scrum Projekt vorschlägt. Dies sichert ab, dass die Scrum Team-Mitglieder für keinen Zeitraum zu viel oder zu wenig Arbeit auf sich nehmen und nicht ihre Zeit und Energie für Arbeit aufwenden, bei der noch wenig Klarheit herrscht.

Einige Vorteile des Time-Boxing sind:

- Ein effizienter Entwicklungsprozess
- Geringere Allgemeine Unkosten
- Hohe Geschwindigkeit für Teams
- Besser fokussierte Teams
- Gut vorbereitete Teammitglieder

Time-Boxing kann in vielen Scrum-Prozessen verwendet werden, darunter beispielsweise bei der Durchführung des täglichen Standup Meetings. Die Dauer des täglichen Standup Meetings ist time-boxed. Mitunter wird Time-boxing verwendet, um exzessives Verbessern eines Ergebnisses zu vermeiden (Vergolden).

Time-Boxing ist eine kritische Praktik innerhalb von Scrum und sollte mit Vorsicht angewandt werden. Arbiträres Time-Boxing kann zur Demotivation des Teams und zur Erzeugung eines beklommenen Arbeitsumfelds führen, weshalb es in angemessener Weise angewandt werden sollte.

2.7.1 Scrum Time-boxes

- **Sprint**— Ein Sprint ist eine Time-Boxed Iteration mit einer Dauer von ein bis sechs Wochen, in welcher der Scrum Master das Scrum Team anleitet, moderiert und von sowohl internen als auch externen Impedimenten während der Erzeugung der zu liefernden Ergebnisse abschirmt. Dies hilft bei der Vermeidung von Fokusverlusten, welche das Ziel des Sprints beeinflussen könnten. In dieser Phase arbeitet das Team daran, die Anforderungen im priorisierten Produkt Backlog in potenziell versandfähige Produkt-Funktionalitäten umzuwandeln. Um den größtmöglichen Nutzen aus einem Scrum-Projekt zu ziehen und ein Höchstmaß an Flexibilität für Änderungen zu gewährleisten, sollte die Dauer eines Sprints so kurz wie möglich sein.

Gleichzeitig muss der Sprint lang genug sein, um das Team in die Lage zu versetzen, ein funktionierendes oder lieferbares Produkt zu erstellen, das vom Product Owner überprüft und genehmigt werden kann..

- **Sprint Planungs Meeting** — Dieses Meeting wird zu Beginn eines jeden Sprints als Teil der Prozesse User Storys Zusagen, Aufgaben Identifizieren, Aufgaben Abschätzen und Sprint Backlog aktualisieren durchgeführt. Es ist auf zwei Stunden pro Woche der Sprintdauer begrenzt. Für einen

einmonatigen/vierwöchigen Sprint sollte die Timebox für ein Sprint Planung-Meeting beispielsweise acht Stunden betragen.

Das Sprint Planungs-Meeting erfüllt die folgenden Ziele:

- Zieldefiniton — Im ersten Teil des Meetings erklärt der Product Owner dem Scrum Team die User Storys oder die Anforderungen im priorisierten Produkt Backlog mit der höchsten Priorität. Das Scrum Team sagt danach in Kollaboration mit dem Product Owner User Storys zu, wodurch das Ziel des Sprints definiert wird.
- Identifikation der Aufgaben — Das Scrum Team legt fest, "wie" die ausgewählten Punkte des priorisierten Produkt Backlogs zu bewältigen sind, um das Ziel des Sprints zu erfüllen.

Die zugesagten User Storys mit ihrem zugehörigen Aufwand werden ins Sprint Backlog eingebracht und dort verfolgt.

- **Tägliches Standup Meeting** — Das tägliche Standup Meeting ist ein kurzes, auf 15 Minuten begrenztes Meeting. Die Teammitglieder treffen sich, um den Fortschritt des Projekts anhand von drei Fragen festzustellen:
 1. Was habe ich seit dem letzten Meeting gemacht?
 2. Was plane ich bis zum nächsten Meeting zu machen?
 3. Vor welchen Impedimenten oder Hemmnissen (falls vorhanden) stehe ich zur Zeit?

Dieses Treffen ist der Kernbestandteil des Prozesses Tägliche Standup Meetings durchführen.

- **Sprint Review Meeting** — Das Sprint Review Meeting ist auf eine Stunde pro Woche der Sprintdauer begrenzt. Für einen einmonatigen/vierwöchigen Sprint sollte die Timebox für ein Sprint Review Meeting beispielsweise vier Stunden betragen. Im Verlaufe des Sprint-Review Meetings, welches als Teil von *Sprint vorführen und validieren* durchgeführt wird, präsentiert das Scrum Team dem Product Owner die zu liefernden Ergebnisse des gegenwärtigen Sprints. Dieser kontrolliert das Produkt (oder das Teilprodukt) im Abgleich mit den Abnahmekriterien und akzeptiert die fertiggestellten User Storys oder lehnt diese ab.
- **Sprint Retrospektions-Meeting** — Das Sprint Retrospektions-Meeting ist auf eine Stunde pro Woche der Sprintdauer begrenzt. Für einen einmonatigen/vierwöchigen Sprint sollte die Timebox für ein Sprint Retrospektions-Meeting beispielsweise vier Stunden betragen. Das Meeting wird im Rahmen von *Sprint Rückblick halten* durchgeführt. Im Verlaufe dieses Meetings trifft sich das Scrum Team, um den vorhergehenden Sprint im Bezug auf die verwendeten Prozesse und Hilfsmittel, Kollaboration und Kommunikationsmechanismen sowie anderen für das Projekt relevanten Aspekten zu überdenken und zu reflektieren. Das Team bespricht, was im vorhergehenden Sprint gelungen und nicht gelungen ist, mit dem Ziel zu lernen und Verbesserungen für die folgenden Sprints zu erreichen. Einige Gelegenheiten zur Verbesserung oder beste Methoden aus diesem Meeting können auch als Teil der Dokumente des Scrum Guidance Body aktualisiert werden.

Abbildung 2-8 zeigt die Time-Boxing-Länge bei Scrum Meetings.

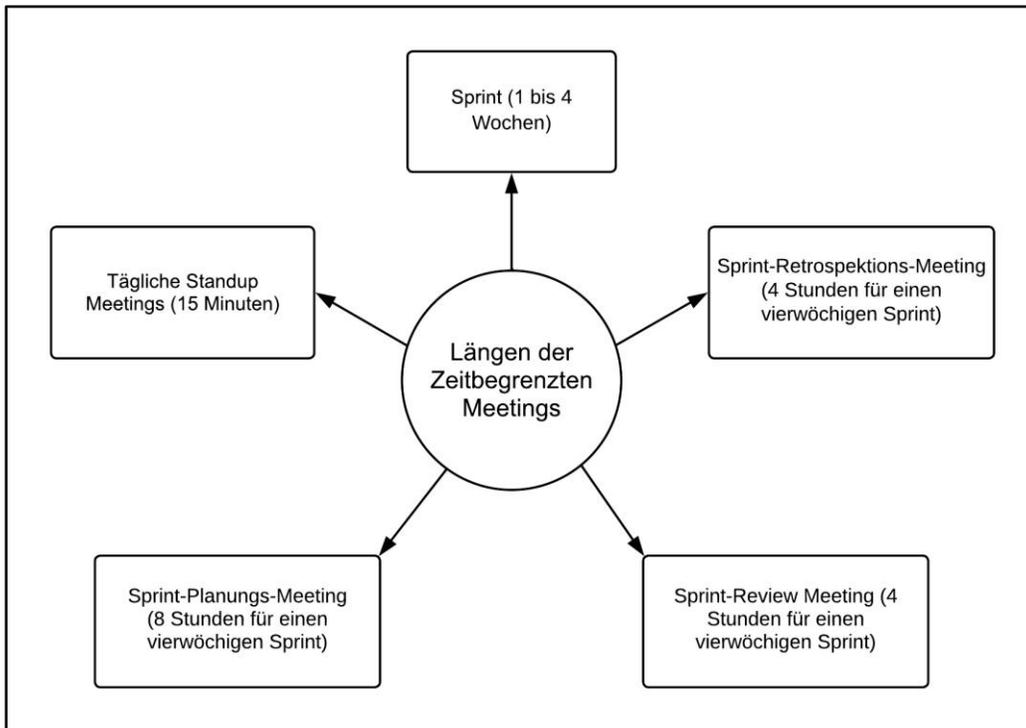


Abbildung 2-8: Time-Boxing-Situationen bei Scrum Meetings

2.8 Iterative Entwicklung

Der Scrum-Rahmen wird von dem Ziel angetrieben, ein Maximum an Geschäftswert in einer möglichst kurzen Zeitspanne zu liefern. Um dies praktisch zu erreichen, glaubt Scrum an die iterative Entwicklung von zu liefernden Ergebnissen.

Bei den meisten komplexen Projekten kann es sein, dass der Kunde nicht in der Lage ist, wirklich konkrete Anforderungen zu definieren, oder sich nicht sicher ist, wie das Endprodukt aussehen soll. Das iterative Modell ist flexibler und stellt dadurch sicher, dass jede Änderung, die vom Kunden gewünscht wird, als Teil des Projekts verwirklicht werden kann. User Storys müssen eventuell kontinuierlich während der gesamten Dauer des Projekts geschrieben werden. In der Frühphase der Formulierung sind die meisten User Storys high-level Funktionalitäten. Diese User Storys werden als Epic(s) bezeichnet. Epic(s) sind üblicherweise zu groß, um durch die Teams in einem einzelnen Sprint fertiggestellt zu werden. Deshalb werden sie in kleinere User Storys unterteilt.

Jeder komplexe Aspekt des Projekts wird während *Priorisiertes Produkt Backlog verfeinern* durch fortwährende Ausarbeitung in kleinere Bestandteile aufgeteilt. *User Storys erstellen* und *User Storys abschätzen, genehmigen und zusagen* werden dazu genutzt, neue Anforderungen zum priorisierten Produkt Backlog hinzuzufügen. Die Aufgabe des Product Owners ist es, einen erhöhten Return of Investment zu sichern, in dem er sich auf den Wert und dessen beständige Lieferung bei jedem Sprint konzentriert. Der Product Owner sollte ein sehr gutes Verständnis der geschäftlichen Rechtfertigung des Projekts und des Wertes, den das Projekt liefern soll, haben, während er das priorisierte Produkt Backlog erstellt und dadurch entscheidet, welche zu liefernden Ergebnisse, und damit welche Werte, in jedem Sprint zu liefern sind. Danach wird während *Aufgaben identifizieren, Aufgaben abschätzen* und *Sprint Backlog aktualisieren* das Sprint Backlog generiert, welches das Team zur Produktion der zu liefernden Ergebnisse nutzt.

Bei jedem Sprint werden dessen Resultate während *Zu liefernde Ergebnisse erzeugen* gewonnen. Der Scrum Master hat sicherzustellen, dass die Scrum Prozesse verwendet werden und leitet das Team so an, dass es auf die produktivste Weise arbeitet. Das Scrum Team organisiert sich selbst und hat das Ziel, anhand der User Storys im Sprint Backlog die zu liefernden Ergebnisse des Sprints zu erzeugen. Bei großen Projekten arbeiten verschiedene funktionsübergreifende Teams während Sprints parallel und erzeugen am Ende jedes Sprints potenziell versandfähige zu liefernde Ergebnisse. Nach Abschluss des Sprints akzeptiert der Product Owner in Sprint vorführen und validieren die zu liefernden Ergebnisse anhand der Abnahmekriterien, oder er lehnt diese ab.

Wie in Abbildung 2-9 aufgezeigt, werden Scrum Projekte auf iterative Weise durchgeführt, indem sie während der gesamten Aktivphase des Projektes Wert liefern.

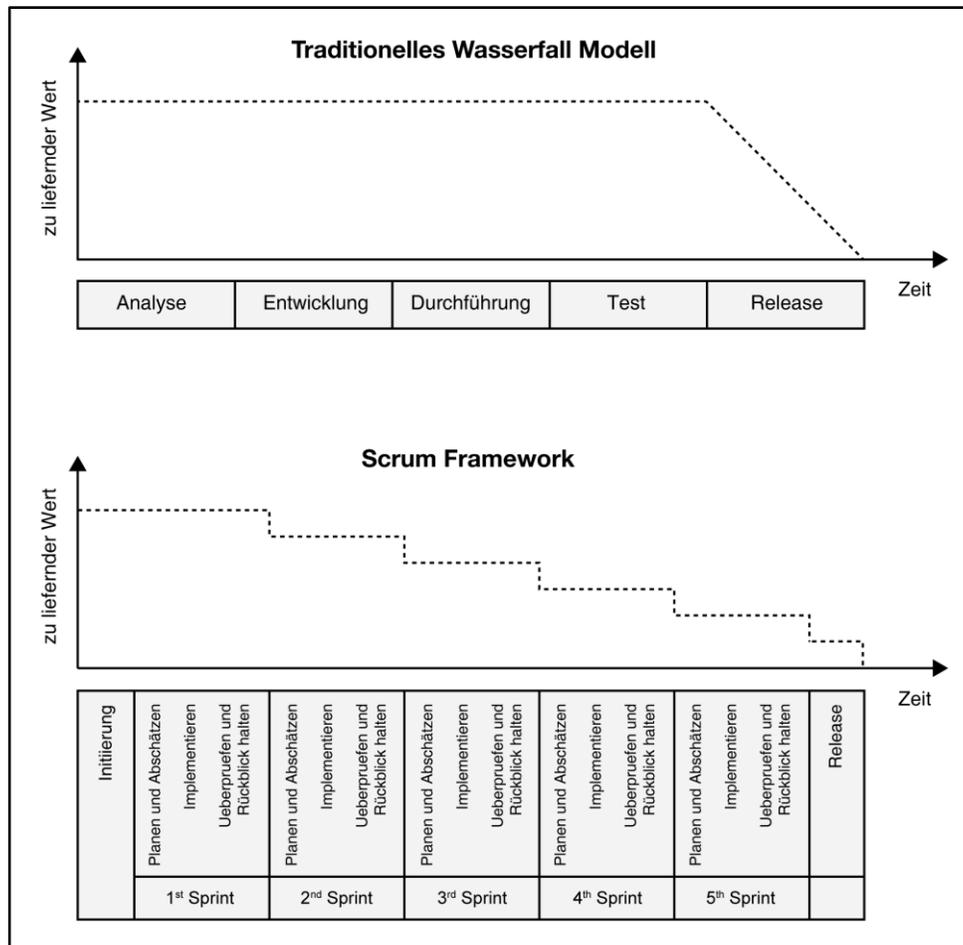


Abbildung 2-9: Scrum vs. Traditionelles Wasserfall Modell

Der Vorteil der iterativen Entwicklung liegt darin, dass sie Kurskorrekturen erlaubt, während alle Beteiligten im Verlaufe des Projekts besser verstehen lernen, was als Teil des Projekts geliefert werden muss und diese Erkenntnisse iterativ in ihre Arbeit einfließen lassen. Dadurch werden Zeit und Aufwand zum Erreichen des Endpunktes deutlich reduziert und das Team produziert zu liefernde Ergebnisse, welche besser auf das leetzendliche Geschäftsumfeld ausgerichtet sind.

2.9 Scrum versus zu traditionelles Projektmanagement

Das Hauptaugenmerk von traditionellem Projektmanagement liegt auf der Durchführung von detaillierten vorausgehenden Planungen für das Projekt mit Gewichtung auf die Fixierung von Umfang, Kosten und Zeitplan sowie das Management dieser Parameter. Traditionelles Projektmanagement kann mitunter zu Situationen führen, in denen der Plan erfolgreich umgesetzt wurde und der Kunde dennoch nicht zufrieden ist.

Der Scrum-Rahmen beruht auf dem Glauben, dass Wissensarbeiter der heutigen Zeit mehr bieten können als nur ihre technische Expertise, und dass das vollständige Ausarbeiten und Planen in einer sich beständig ändernden Umgebung nicht effizient ist. Daher unterstützt Scrum eine faktenbasierte, iterative Entscheidungsfindung. Bei Scrum liegt das Hauptaugenmerk auf der Lieferung eines Produkts, das die Anforderungen des Kunden erfüllt.

Um den größtmöglichen Mehrwert in der kürzestmöglichen Zeit zu liefern, fördert Scrum Priorisierung und Time-Boxing anstatt Umfang, Kosten und Zeitplan des Projekts zu fixieren. Eine wichtige Funktion von Scrum ist Selbstorganisation, welche es den Individuen, die tatsächlich die Arbeit leisten, ermöglicht, die Aufgaben abzuschätzen und die Verantwortung für die Aufgaben zu übernehmen.

3. ORGANISATION

3.1 Einleitung

In diesem Abschnitt werden die verschiedenen Facetten einer Scrum Projekt Organisation sowie Kernrollen und nicht-zentrale Rollen besprochen und es wird erklärt, wie ein Scrum Hochleistungs-Team zusammengestellt wird.

Organisation, wie im *Scrum Body of Knowledge (SBOK®)* definiert, ist auf Folgendes anwendbar:

- Portfolios, Programme und/oder Projekte in jeder Branche
- Produkte, Dienstleistungen oder jegliche an Business Stakeholdern zu liefernde Ergebnisse
- Projekte jeglicher Größe oder Komplexität

Der Begriff "Produkt" in dieser Wissenssammlung kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann auf jedes Projekt in jeder Branche effektiv angewendet werden — von kleinen Projekten oder Teams mit lediglich sechs Team-Mitgliedern bis hin zu großen, komplexen Projekten mit mehreren Hundert Team-Mitgliedern.

Dieses Kapitel ist in folgende Abschnitte aufgeteilt:

3.2 Leitfaden für die Kernrollen—Dieser Abschnitt klärt, welche Abschnitte oder Unterabschnitte wichtig für den Product Owner, Scrum Master und das Scrum Team sind.

3.3 Scrum Projektrollen—Dieser Abschnitt behandelt alle mit dem Scrum Projekt verbundenen Kernrollen und nicht-zentrale Rollen.

3.4 Product Owner —Dieser Abschnitt stellt die Hauptverantwortlichkeiten des Product Owners in Bezug auf ein Scrum Projekt, **Programmen und Portfolios** heraus.

3.5 Scrum Master —Dieser Abschnitt konzentriert sich auf die Hauptverantwortlichkeiten des Scrum Masters im Rahmen eines Scrum Projekts, **Programmen und Portfolios**.

3.6 Scrum Team —Dieser Abschnitt betont die Hauptverantwortlichkeiten des Scrum Teams in einem Scrum Projekt.

3.7 Scrum in Projekten, Programmen und Portfolios—Der Schwerpunkt dieses Abschnitts liegt auf dem Zuschnitt des Scrum Rahmens auf und seine Benutzung in verschiedenen Kontexten von Programmen und Portfolios. Auch werden spezifische Verantwortlichkeiten der Scrum Team-Mitglieder in Bezug auf Kommunikation, Integration und der gemeinschaftlichen Arbeit mit den Programm Management Teams hervorgehoben.

3.8 Verantwortlichkeiten—Dieser Abschnitt beschreibt die zum Thema Organisation relevanten Verantwortlichkeiten für jeden am Projekt Arbeitenden, basierend auf dessen Rolle.

3.9 Scrum versus traditionelles Projekt Management —Dieser Abschnitt erklärt die Hauptunterschiede und Vorteile des Scrum Modells gegenüber den traditionellen Wasserfall Modellen des Projekt Management.

3.10 Populäre Theorien aus dem Personalbereich und deren Relevanz in Scrum—Dieser Abschnitt enthält einige der populärsten Theorien aus dem Personalbereich, die für alle Mitglieder des Scrum Kernteams nützlich sind.

3.2 Leitfaden für die Kernrollen

1. **Product Owner** —Es ist zwingend erforderlich für den Product Owner, das gesamte Kapitel zu lesen.
2. **Scrum Master** —Der Scrum Master sollte mit diesem kompletten Kapitel vertraut sein, besonders mit den Abschnitten 3.5, 3.6, 3.7, 3.8, und 3.9.
3. **Scrum Team** —Das Scrum Team sollte sich hauptsächlich auf die Abschnitte 3.6, 3.7, und 3.8 konzentrieren.

3.3 Scrum Projekt Rollen

Um die erfolgreiche Implementierung eines Scrum Projekts zu gewährleisten, ist es sehr wichtig, die definierten Rollen und Verantwortlichkeiten zu verstehen.

Scrum Rollen werden in zwei ausgedehnte Kategorien gegliedert:

1. **Kernrollen**—Kernrollen sind die Rollen, welche zwingend für die Erstellung des Produkts eines Projekts benötigt werden, sind an das Projekt gebunden und schlussendlich für den Erfolg eines jeden Sprints sowie des gesamten Projekts verantwortlich.
2. **Nicht-zentrale Rollen**—Nicht-zentrale Rollen sind solche Rollen, die nicht zwingend für das Scrum-Projekt erforderlich sind. Sie können Teammitglieder beinhalten, die Interesse am Projekt haben, aber keine formale Rolle im Projektteam spielen. Diese Rollen mögen eine Schnittstelle zum Team haben, sind jedoch nicht für den Erfolg des Projekts verantwortlich. Nicht-zentrale Rollen sollten in jedem Scrum Projekt in Betracht gezogen werden.

3.3.1 Kernrollen

Es gibt drei Kernrollen in Scrum, die schlussendlich dafür verantwortlich sind, die Projektziele zu erreichen. Diese Kernrollen sind der Product Owner, der Scrum Master und das Scrum Team. Zusammen werden sie als das Scrum Kernteam bezeichnet. Besonders zu erwähnen ist, dass keine dieser drei Kernrollen Verfügungsgewalt über die anderen hat.

1. Product Owner

Der Product Owner ist die verantwortliche Person für die Maximierung des Geschäftswert im Projekt. Er/sie trägt die Verantwortung, Kundenanforderungen zu artikulieren und die geschäftliche Rechtfertigung des Projekts zu erhalten. Der Product Owner repräsentiert die *Stimme des Kunden*.

Der Rolle des Product Owners in einem Projekt entsprechend kann es einen Programm Product Owner in einem Programm oder einen Portfolio Product Owner in einem Portfolio geben.

2. Scrum Master

Der Scrum Master stellt sicher, dass dem Team ein Umfeld zur Verfügung steht, das die erfolgreiche Fertigstellung der Produktentwicklung fördert. Der Scrum Master leitet an, unterstützt und lehrt jeden, der ins Projekt involviert ist, Scrum Praktiken; beseitigt Hindernisse für das Team; und stellt sicher dass Scrum Prozesse befolgt werden.

Zu beachten ist, dass die Rolle des Scrum Masters sehr unterschiedlich von der Rolle ist, die der Projekt Manager im Projekt Management des traditionellen Wasserfall Modells spielt, in dem der Projekt Manager als Manager oder Projektleiter arbeitet. Der Scrum Master arbeitet nur als Unterstützer und er/sie steht auf der gleichen hierarchischen Ebene wie jeder andere im Scrum Team — jede Person des Scrum Teams, die lernt, Scrum Projekte zu moderieren, kann Scrum Master für ein Projekt oder einen Sprint werden.

Der Rolle des Scrum Masters in einem Projekt entsprechend, kann es einen Programm Scrum Master in einem Programm oder einen Portfolio Scrum Master in einem Portfolio geben.

3. Scrum Team

Das Scrum Team ist eine Gruppe oder ein Team von Leuten, deren Verantwortung es ist, die vom Product Owner festgelegten geschäftlichen Anforderungen zu verstehen, User Storys abzuschätzen und die endgültigen zu liefernden Ergebnisse zu erstellen.

Abbildung 3-1 gibt einen Überblick der Scrum Kernteam Rollen.

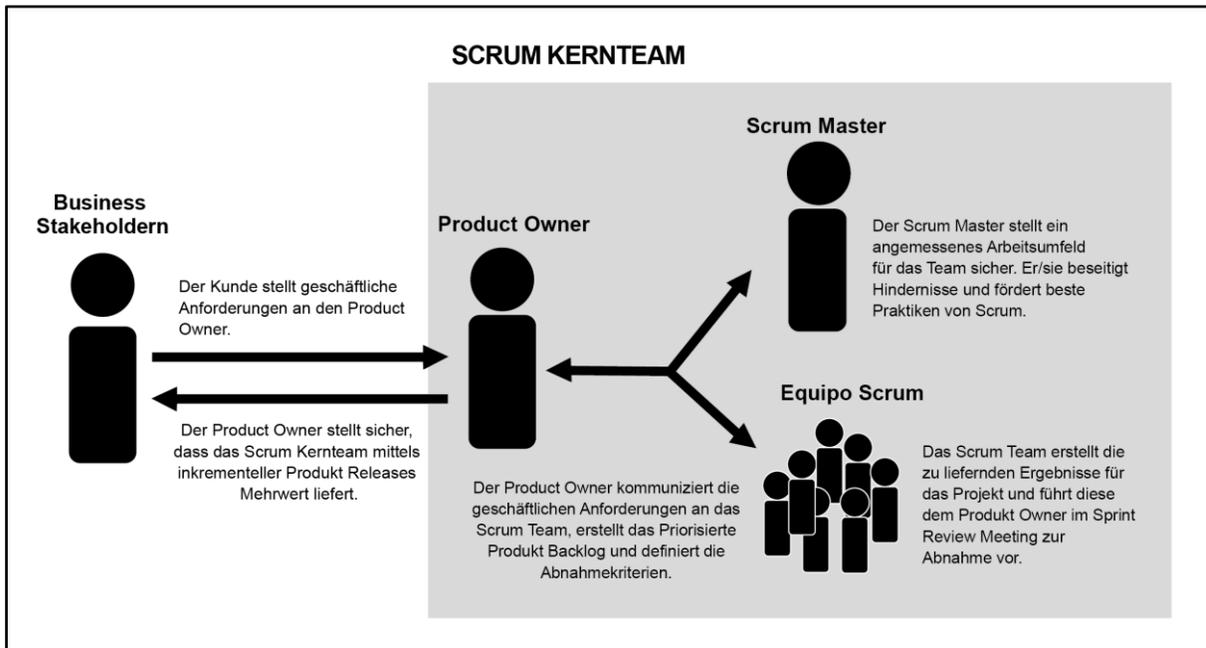


Abbildung 3-1: Scrum Rollen—Überblick

3.3.2 Nicht-zentrale Rollen

Die nicht-zentralen Rollen sind die Rollen, welche nicht zwingend für das Scrum Projekt vorgeschrieben sind und nicht kontinuierlich oder direkt in die Scrum Prozesse involviert sind. Trotzdem ist es wichtig, die nicht-zentralen Rollen zu kennen, denn in manchen Scrum Projekten können sie eine wesentliche Rolle spielen.

Nicht-zentrale Rollen können folgende Rollen einschließen:

1. Business Stakeholdern

Business Stakeholdern schließt als umfassender Begriff Kunden, Anwender und Sponsoren ein, die regelmäßig mit Product Owner, Scrum Master und Scrum Team interagieren, um Input zu liefern und die Schaffung des mit dem Projekt verbundenen Produkts, Dienstleistung oder anderer Ergebnisse zu unterstützen. Business Stakeholdern beeinflussen das Projekt die ganze Projektentwicklung hindurch. Business Stakeholdern können auch während *Epics entwickeln*, *Priorisiertes Produkt Backlog erstellen*, *Release-Planung durchführen*, *Sprint Rückblick halten* und *anderen wichtigen Prozessen* in Scrum eine Rolle spielen. Scrum verlangt von den Business Stakeholdernn komplette Unterstützung.

Die Verantwortung für die Einbindung der Business Stakeholdern liegt beim Product Owner. Die folgenden Maßnahmen werden empfohlen, um das Engagement und die Unterstützung der Business Stakeholdern aufrechtzuerhalten:

- Sicherstellung einer effektiven Zusammenarbeit und Einbindung der Business Stakeholdern in das Projekt
- Kontinuierliche Bewertung der geschäftlichen Auswirkungen
- Aufrechterhaltung einer regelmäßigen Kommunikation mit den Business Stakeholdern
- Steuerung der Erwartungen der Business Stakeholdern

- **Kunde**

Beim Kunden handelt es sich um eine Einzelperson oder eine Organisation, die das/den aus dem Projekt entstandene Produkt, Service oder generelle Resultat erwirbt. Für jede Organisation kann es je nach Projekt sowohl interne Kunden (d.h. innerhalb derselben Organisation) als auch externe Kunden (d.h. außerhalb der Organisation) geben.

- **Anwender**

Anwender sind die Personen oder die Organisation, die Produkt, Service oder andere Ergebnisse des Projekts direkt benutzen. Ähnlich zu Kunden, können Anwender sowohl organisationsinterne als auch externe Anwender sein. In einigen Fällen können Kunde und Anwender derselbe sein.

- **Sponsor**

Der Sponsor ist eine individuelle Person oder eine Organisation, der Mittel und Unterstützung für das Projekt zur Verfügung stellt. Der Sponsor ist außerdem der Business Stakeholdern dem letztendlich alle Rechenschaft ablegen.

Der Sponsor sollte das finanzielle Endergebnis im Zusammenhang mit einem Produkt oder einer Dienstleistung verstehen und ist in der Regel eher an den Endergebnissen als an einzelnen Aufgaben interessiert. Es ist wichtig, dass der Sponsor (oder die Sponsoren), die das Projekt finanzieren, Klarheit über die folgenden Überlegungen haben:

- Vorteile der Anwendung von Scrum-Praktiken für das Projekt
- Zieltermine und geschätzte Kosten des Scrum-Projekts
- Gesamtrisiken, die mit dem Scrum-Projekt verbunden sind, und die Schritte, um sie zu mindern oder zu vermeiden
- Erwartete Freigabetermine und endgültige zu Liefernde Ergebnisse

2. Unterstützende Dienste

Unterstützende Dienste sind interne oder externe Gruppen, die das Scrum-Projekt unterstützen oder von ihm betroffen sind, z. B. Schulung, Logistik, Marketing, Finanzen, Infrastruktur usw.

3. Anbieter

Anbieter umfassen externe Personen oder Organisationen, die Produkte und Dienstleistungen bereitstellen, die nicht innerhalb der Kernkompetenzen der Projektorganisation liegen.

4. Scrum Guidance Body

Der Scrum Guidance Body (SGB) ist eine optionale Rolle die allerdings unbedingt zu empfehlen ist, um den praktischen Umgang der Organisation mit Scrum zu formalisieren. Er besteht in der Regel aus einer Reihe von Dokumenten und/oder einer Gruppe von Experten, die in der Regel in die Festlegung der Ziele hinsichtlich Qualität, staatlichen Vorschriften, Sicherheit und anderer organisatorischer Schlüsselparameter einbezogen sind. Diese Ziele steuern die Arbeit von Product Owner, Scrum Master und dem Scrum Team. Der Scrum Guidance Body hilft außerdem, die besten Praktiken aufzugreifen, die in allen Scrum Projekten der Organisation benutzt werden sollten.

Der Scrum Guidance Body trifft keine Entscheidungen, die das Projekt betreffen. Stattdessen agiert er in einer Beratungs-oder Anleitungs-Funktion für alle hierarchischen Ebenen der Projektorganisation—Portfolio, Programm und Projekt. Scrum Teams haben die Möglichkeit, den Scrum Guidance Body nach Bedarf um Rat zu fragen.

3.4 Product Owner

Der Product Owner vertritt die Interessen der Business Stakeholdern Gemeinschaft vor dem Scrum Team. Es liegt in der Verantwortung des Product Owners sicherzustellen, dass Funktionalitätsanforderungen an das Produkt oder die Dienstleistung verständlich an das Scrum Team kommuniziert, sowie Abnahmekriterien definiert und erfüllt werden. In anderen Worten ist der Product Owner dafür verantwortlich, dass das Scrum Team Werte liefert. Der Product Owner muss immer zwei Standpunkte im Auge haben. Er/sie muss die Interessen und Befürnisse aller Business Stakeholdern verstehen und unterstützen, gleichzeitig aber auch die Bedürfnisse und Tätigkeiten des Scrum Teams verstehen. Da der Product Owner die Bedürfnisse und Prioritäten der Business Stakeholdern, einschließlich Kunden und Anwender, verstehen muss, wird diese Rollen meist als Stimme des Kunden bezeichnet.

Tabelle 3-1 fasst die Verantwortlichkeiten des Product Owners in den verschiedenen Scrum Prozessen zusammen.

Prozess	Verantwortlichkeiten des Product Owners
8.1 Projekt Vision erstellen	<ul style="list-style-type: none"> • Definiert die Projekt Vision • Hilft den Projektauftrag und das Projektbudget zu erstellen
8.2 Scrum Master und Business Stakeholdern identifizieren	<ul style="list-style-type: none"> • Hilft den Scrum Master für das Projekt festzulegen • Identifiziert Business Stakeholdern
8.3 Scrum Team bilden	<ul style="list-style-type: none"> • Hilft die Scrum Team-Mitglieder zu bestimmen • Hilft einen Kollaborationsplan zu erstellen • Erstellt mit dem Scrum Master den Team-Building Plan
8.4 Epics entwickeln	<ul style="list-style-type: none"> • Entwickelt Epics und Personas
8.5 Priorisiertes Produkt Backlog erstellen	<ul style="list-style-type: none"> • Priorisiert Einträge im Priorisierten Produkt Backlog • Definiert Done-Kriterien und erfüllt die Definition von Ready
8.6 Release-Planung durchführen	<ul style="list-style-type: none"> • Erstellt den Release Plan • Hilft die Sprint-Länge festzulegen
9.1 User Storys Erstellen	<ul style="list-style-type: none"> • Verantwortlich für die Erstellung von User Storys • Definiert Abnahmekriterien für jede User Story
9.2 User Storys Abschätzen	<ul style="list-style-type: none"> • Genehmigt User Storys
9.3 User Storys Zusagen	<ul style="list-style-type: none"> • Unterstützt das Scrum Team und dessen Zusagen für User Storys
9.4 Aufgaben Identifizieren	<ul style="list-style-type: none"> • Erklärt dem Scrum Team User Storys, während dieses die Aufgabenliste erstellt
9.5 Aufgaben abschätzen	<ul style="list-style-type: none"> • Bietet Anleitung und Erläuterungen für das Scrum Team bei der Abschätzung des Aufwands jeder Aufgabe
9.6 Sprint Backlog aktualisieren	<ul style="list-style-type: none"> • Erklärt dem Scrum Team die Anforderungen, während dieses das Sprint Backlog aktualisiert
10.1 Zu liefernde Ergebnisse erzeugen	<ul style="list-style-type: none"> • Verdeutlicht dem Scrum Team Geschäftliche Anforderungen
10.3 Priorisiertes Produkt Backlog verfeinern	<ul style="list-style-type: none"> • Verfeinert das Priorisierte Produkt Backlog
11.1 Sprint Vorführen und Validieren	<ul style="list-style-type: none"> • Akzeptierung/Ablehnung zu liefernder Ergebnisse • Liefert dem Scrum Master und dem Scrum Team notwendiges Feedback • Aktualisiert den Release Plan und das Priorisierte Produkt Backlog
12.1 Zu liefernde Ergebnisse ausliefern	<ul style="list-style-type: none"> • Hilft bei der Bereitstellung von Produkt-Releases und koordiniert diese mit dem Kunden
12.2 Release Rückblick halten	<ul style="list-style-type: none"> • Nimmt an Release Rückblick Meetings teil

Tabelle 3-1: Verantwortlichkeiten des Product Owners in Scrum Prozessen

Die anderen Verantwortlichkeiten des Product Owners sind:

- Die gesamten Anfangsanforderungen des Projekts bestimmen und die Projekt Tätigkeiten anstoßen; dies mag Interaktion mit dem Programm Product Owner und dem Portfolio Product Owner einschließen, um sicherzustellen, dass das Projekt mit den Anweisungen des Senior Managements im Einklang ist
- Anwender des Produkts oder der Dienstleistung mithilfe eines tiefen Verständnisses der Anwendergemeinschaft repräsentieren
- Die anfängliche und laufende Sicherung der Finanzressourcen für das Projekt
- Konzentration auf Werterstellung und gesamte Investitionsrendite (ROI)
- Die Durchführbarkeit beurteilen und die Lieferung des Produkts oder der Dienstleistung gewährleisten

Der Product Owner muss nicht immer einen externen Kunden oder ein externes Unternehmen vertreten. Bei einem IT-Projekt können beispielsweise Anforderungen wie die Verbesserung der Leistung, Skalierbarkeit, Testbarkeit, Zuverlässigkeit, Informationssicherheit und Konformität von den Technologiegruppen innerhalb des Unternehmens gestellt werden. Product Owner können in solchen Fällen auch Rollen wie technische Architekten, technische Leiter usw. haben.

3.4.1 Stimme des Kunden (VOC)

Als Repräsentant des Kunden und anderer Business Stakeholdern ist der Product Owner die sogenannte Stimme des Kunden (Voice of the Customer) und stellt sicher, dass die expliziten und impliziten Bedürfnisse des Kunden in User Storys im Priorisierten Produkt Backlogs übersetzt und später benutzt werden, um die zu liefernde Ergebnisse für den Kunden zu erstellen.

3.5 Scrum Master

Der Scrum Master ist der „Unterstützende Leader“ des Scrum Teams, er moderiert und unterstützt Team Interaktionen als Coach und Motivator. Der Scrum Master hat die Verantwortung sicherzustellen, dass das Team ein produktives Arbeitsumfeld hat, indem er das Team vor externen Einflüssen schützt, Hindernisse beseitigt und Scrum Prinzipien, Aspekte und Prozesse durchsetzt.

Tabelle 3-2 fasst die Verantwortlichkeiten des Scrum Masters in verschiedenen Scrum Prozessen zusammen.

Prozesse	Verantwortlichkeiten des Scrum Masters
8.2 Scrum Master und Business Stakeholdern identifizieren	<ul style="list-style-type: none"> • Hilft den/die Business Stakeholdern für das Projekt zu identifizieren
8.3 Scrum Team bilden	<ul style="list-style-type: none"> • Unterstützt die Auswahl des Scrum Teams • Unterstützt die Erstellung des Kollaborationsplans und des Team-Building Plans • Stellt sicher, dass Backup Ressourcen für einen reibungslosen Projektablauf zur Verfügung stehen
8.4 Epics entwickeln	<ul style="list-style-type: none"> • Unterstützt die Entwicklung von Epics und Personas
8.5 Priorisiertes Produkt Backlog erstellen	<ul style="list-style-type: none"> • Hilft dem Product Owner bei der Erstellung des Priorisierten Produkt Backlogs, der Definition der Done-Kriterien, und bei der Erfüllung der Definition von Ready
8.6 Release Planung durchführen	<ul style="list-style-type: none"> • Koordiniert die Erstellung des Release Plans • Hilft dem Product Owner und dem Scrum Team die Sprint-Länge festzulegen
9.1 User Storys Erstellen	<ul style="list-style-type: none"> • Unterstützt die Erstellung von User Storys und deren Abnahmekriterien
9.2 User Storys abschätzen	<ul style="list-style-type: none"> • Unterstützt meetings des Scrum Teams zur abschätzen von User Storys
9.3 User Storys Zusagen	<ul style="list-style-type: none"> • Organisiert und moderiert Meetings des Scrum Teams zur zusagen von User Storys
9.4 Aufgaben identifizieren	<ul style="list-style-type: none"> • Unterstützt das Scrum Team in der Erstellung der Aufgabenliste für den nächsten Sprint
9.5 Aufgaben abschätzen	<ul style="list-style-type: none"> • Unterstützt das Scrum Team bei der Abschätzung des zur Erfüllung der für den Sprint festgesetzten Aufgaben benötigten Arbeitsaufwands
9.6 Sprint Backlog aktualisieren	<ul style="list-style-type: none"> • Unterstützt das Scrum Team bei der Aktualisierung des Sprint Backlogs und der Erstellung des Sprint Burndown Charts
10.1 Zu liefernde Ergebnisse erzeugen	<ul style="list-style-type: none"> • Unterstützt das Scrum Team in der Erzeugung der für den Sprint festgelegten zu liefernden Ergebnisse • Hilft das Scrumboard und das Impediment Log zu aktualisieren
10.2 Tägliches Standup durchführen	<ul style="list-style-type: none"> • Stellt sicher, dass das Scrumboard und das Impediment Log aktuell bleiben
10.3 Priorisiertes Produkt Backlog verfeinern	<ul style="list-style-type: none"> • Unterstützt Priorisiertes Produkt Backlog Review Meetings
11.1 Zum Scrum der Scrums zusammenkommen	<ul style="list-style-type: none"> • Stellt sicher, dass Probleme, die das Scrum Team betreffen, besprochen und gelöst werden
11.2 Sprint Vorführen und Validieren	<ul style="list-style-type: none"> • Organisiert und moderiert Vorführungen der fertiggestellten zu liefernden Ergebnisse des Scrum Teams, um die Genehmigung des Product Owners zu erhalten
11.3 Sprint Rückblick halten	<ul style="list-style-type: none"> • Stellt sicher, dass in den nachfolgenden Sprints ein ideales Projekt Umfeld für das Scrum Team vorhanden ist
12.2 Release Rückblick halten	<ul style="list-style-type: none"> • Vertritt das Scrum Kernteam, um, falls nötig, Lehren aus dem laufenden Projekt bereitzustellen

Tabelle 3-2: Verantwortlichkeiten des Scrum Masters in Scrum Prozessen

3.6 Scrum Team

Das Scrum Team wird manchmal auch als Entwicklungsteam bezeichnet, da es für die Entwicklung des Produkts, der Dienstleistung oder anderer Ergebnisse verantwortlich ist. Es besteht aus einer Gruppe von selbstorganisierten Personen, die an den User Storys im Sprint Backlog arbeiten um die zu liefernden Ergebnisse des Projekts zu erstellen.

Tabelle 3-3 fasst die Verantwortlichkeiten des Scrum Teams in den verschiedenen Scrum Prozessen

Prozesse	Verantwortlichkeiten des Scrum Teams
8.3 Scrum Team bilden	<ul style="list-style-type: none"> Bietet Input zur Erstellung des Kollaborationsplans und des Team-Building Plans
8.4 Epics entwickeln	<ul style="list-style-type: none"> Stellt ein klares Verständnis der Epics und Personas sicher
8.5 Priorisiertes Produkt Backlog erstellen	<ul style="list-style-type: none"> Versteht die User Storys im Priorisierten Produkt Backlog
8.6 Release-Planung durchführen	<ul style="list-style-type: none"> Einigt sich mit anderen Mitgliedern des Scrum Kernteams über die Sprint-Länge Ersucht Klarstellungen über neue Produkte oder Änderungen der bestehenden Produkte, falls vorhanden, im verfeinerten Priorisierten Produkt Backlog
9.1 User Storys Erstellen	<ul style="list-style-type: none"> Liefert dem Product Owner Input zu den erstellten User Storys
9.2 User Storys abschätzeng	<ul style="list-style-type: none"> Abschätzung der vom Product Owner genehmigten User Storys
9.3 User Storys zusagen	<ul style="list-style-type: none"> Zusage die User Storys in einem Sprint fertig zu stellen
9.4 Aufgaben identifizieren	<ul style="list-style-type: none"> Erstellen einer Aufgabenliste, basierend auf den zugesagten User Storys und Abhängigkeiten
9.5 Aufgabenabschätzung	<ul style="list-style-type: none"> Geschätzten Arbeitsaufwand einer Aufgabe identifizieren und falls nötig, aktualisieren der Aufgabenlist
9.6 Sprint Backlog aktualisieren	<ul style="list-style-type: none"> Definiert die User Storys und Aufgaben, die in das Sprint Backlog aufgenommen und im Sprint Burndown Chart verfolgt werden sollen
10.1 Zu liefernde Ergebnisse erzeugen	<ul style="list-style-type: none"> Erzeugt die zu liefernden Ergebnisse Identifiziert Risiken und setzt Maßnahmen zur Risikominderung um Identifiziert die Hindernisse, die im Hindernis Log verfolgt werden sollen
10.2 Tägliches Standup durchführen	<ul style="list-style-type: none"> Scrumboard während jedes Sprints aktualisieren Probleme einzelner Mitglieder besprechen und Lösungen zur Motivation des Teams suchen
10.3 Priorisiertes Produkt Backlog verfeinern	<ul style="list-style-type: none"> Teilnahme an Priorisiertes Produkt Backlog Rückblick Meetings
11.1 Sprint Vorführen und Validieren	<ul style="list-style-type: none"> Führt dem Product Owner die fertiggestellten zu liefernden Ergebnisse zur Genehmigung vor
11.2 Sprint Rückblick halten	<ul style="list-style-type: none"> Identifiziert Verbesserungsmöglichkeiten aus dem laufenden Sprint und einigt sich auf umsetzbare Verbesserungen für den nächsten Sprint
12.2 Release Rückblick halten	<ul style="list-style-type: none"> Teilnahme am Release Rückblick Meeting

Tabelle 3-3: Verantwortlichkeiten des Scrum Teams in den Scrum Prozessen

3.6.1 Wahl der Mitarbeiter

Abbildung 3-3 zählt wünschenswerte Eigenschaften für die Scrum Kernrollen auf

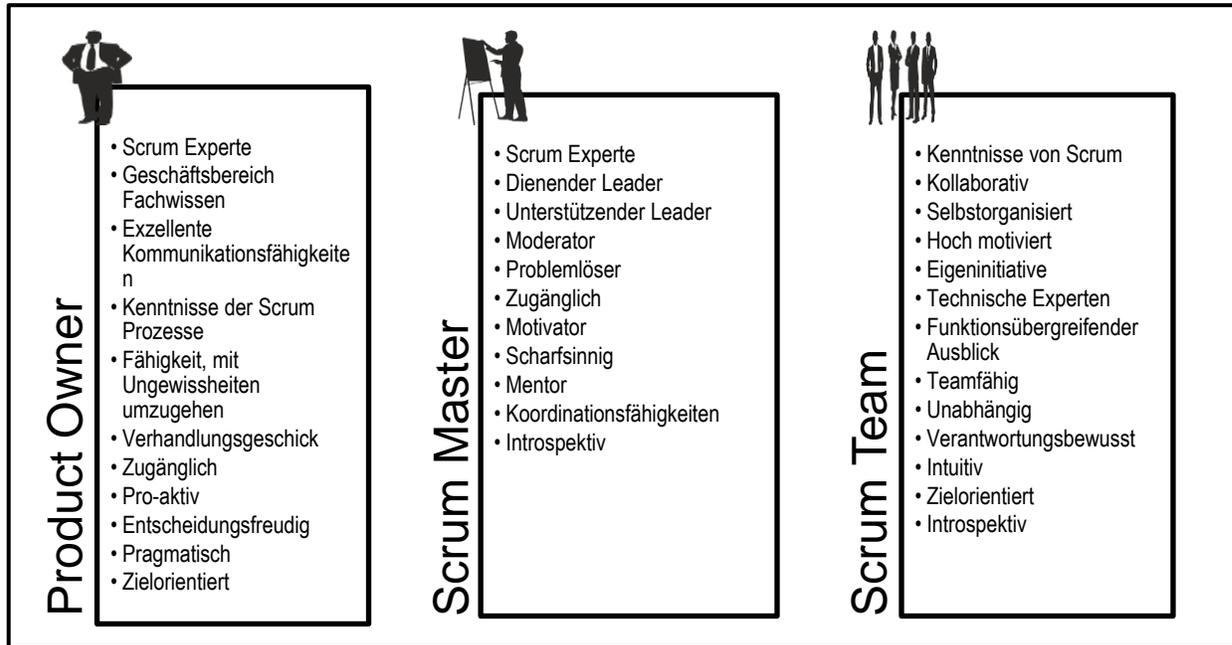


Abbildung 3-2: Wünschenswerte Eigenschaften für die Scrum Kernrollen

3.6.2 Die Größe des Scrum Teams

Es ist sehr wichtig, alle entscheidenden Fähigkeiten im Scrum Team zu besitzen, die zur Durchführung der Arbeit am Projekt benötigt werden. Auch ist es notwendig, ein hohes Mass an Zusammenarbeit zu haben, um die Produktivität zu maximieren, sodass nur minimale Koordination benötigt wird, um Dinge zu erledigen.

Die optimale Größe eines Scrum Teams beträgt sechs bis zehn Mitglieder—groß genug um eine adäquate Zusammenstellung an Fähigkeiten zu gewährleisten, aber auch klein genug, um einfach zusammenzuarbeiten. Das Ziel ist, genug Mitglieder im Team zu haben, dass die Arbeit erledigt wird, das Team aber klein genug zu halten, dass effektive Kommunikation und Management möglich bleiben. Allerdings gibt es auch Nachteile von kleineren Teams. Ein großer Nachteil ist, dass kleinere Teams vom Verlust eines Mitgliedes, selbst für eine kurze Zeit, schwerer betroffen sind. Um dieses Problem anzugehen, könnte man Teammitglieder mit Fachwissen und Fähigkeiten außerhalb ihrer eigenen spezifischen Rollen auswählen. Jedoch mag dies schwierig sein und von der Art des Projekts, der Branche und Größe der Organisation abhängen. Es wird daher empfohlen, Reservekandidaten zu haben, um jede Person, die das Scrum Team verlassen muss, ersetzen zu können.

3.7 Scrum in Großen Projekten, Programmen und Portfolios

3.7.1 Wie Scrum in einem großen Projekt funktioniert

Die in den Kapiteln 8 bis 12 definierten grundlegenden Scrum-Prozesse gelten für Scrum-Projekte mit einem Product Owner, einem Scrum Master und einem bis drei Scrum-Teams. Diese werden normalerweise als kleine Scrum-Projekte angesehen.

Bei großen Projekten, die den Einsatz von vier oder mehr Scrum-Teams mit mehreren Product Ownern und mehreren Scrum-Mastern erfordern, bleiben die in den Kapiteln 8 bis 12 definierten grundlegenden Prozesse gültig, aber einige zusätzliche Überlegungen und Aktualisierungen der Inputs, Hilfsmittel und Outputs können erforderlich sein. Dies kann zusätzliche Koordinations- und Synchronisationsaufwendungen beinhalten. Die Auswirkungen auf die grundlegenden Scrum-Prozesse bei der Skalierung von Scrum auf große Projekte werden in Kapitel 13 ausführlich beschrieben.

Die Definition, was ein großes Projekt ist, hängt in der Regel von der Organisation und/oder der Komplexität der Projekte ab, die durchgeführt werden. Ein Schlüsselkriterium dafür, ob ein Projekt als klein oder groß eingestuft wird, ist, ob das Projekt mehrere Scrum Master und/oder mehrere Product Owner erfordert. Wenn das Projekt nur einen Scrum Master und einen Product Owner erfordert, dann können diese Personen normalerweise alle zusätzlichen Kommunikations- und Synchronisationsanforderungen, die das Projekt mit sich bringt, bewältigen.

Einige Gründe, warum bei großen Projekten zusätzliche Inputs, Hilfsmittel und Outputs benötigt werden, sind folgende:

Product Owner

- Notwendigkeit der Zusammenarbeit zwischen Product Ownern bei der Arbeit mit Business Stakeholdern, der Verfeinerung des priorisierten Product Backlogs und der Arbeit mit mehreren Scrum Teams

Es ist auch wichtig zu beachten, dass bei der Skalierung von Scrum für große Projekte zusätzliche unterstützende Dienste wie Architekten, Produktmanager, Compliance, Informationssicherheit, Governance-Gremien usw. benötigt werden können.

Scrum Master

- Notwendigkeit der Zusammenarbeit zwischen Scrum Masters bei der Beseitigung von Hindernissen und bei der Synchronisierung der Arbeit mehrerer Scrum Teams

Scrum Teams

- Zunehmende Interaktion und Abhängigkeiten zwischen Scrum-Teams bei zunehmender Komplexität eines großen Projekts
- Notwendigkeit, Konflikte zu managen, Probleme zu lösen und Prioritäten zwischen den einzelnen Scrum Teams zu setzen
- Erfordernis der Spezialisierung, da einige Scrum Teams spezialisierte Ressourcen für bestimmte Aufgaben benötigen (und diese besonderen Fertigkeiten nicht in allen Scrum Teams benötigt werden)

- Notwendigkeit, bestimmte Richtlinien und Standards zu definieren, die von allen Scrum Teams eingehalten werden sollten (z.B. Sicherheitsstandards innerhalb eines Unternehmens oder gesetzliche und behördliche Richtlinien für bestimmte Branchen); diese können vom Scrum Guidance Body definiert werden
- Erfordernis, eine Umgebung oder einen Arbeitsbereich für das große Projekt einzurichten, der dann von allen Scrum Teams genutzt wird
- Notwendigkeit, die Ergebnisse mehrerer Scrum-Teams zu koordinieren, um die Freigabe eines großen Projekts zu erleichtern.

3.7.2 Zusätzliche Kernrollen in einem großen Projekt

Al escalar Scrum en grandes proyectos, se necesitan los siguientes roles principales:

3.7.2.1 Chief Product Owner

Im Falle von großen Projekten mit zahlreichen Scrum Teams und mehreren Product Ownern ist es weiterhin notwendig, eine einzelne Person zu haben, die Tag für Tag die geschäftlichen Entscheidungen trifft. Diese Person/Rolle ist der Chief Product Owner. Diese Rolle ist dafür verantwortlich, die Arbeit von mehreren Product Ownern zu koordinieren. Mit Hilfe der Product Owner bereitet der Chief Product Owner das gesamte Priorisierte Produkt Backlog vor und pflegt es für das große Projekt. Er verwendet es, um die Arbeit über die Product Owner der Scrum Teams zu koordinieren. Der Chief Product Owner ist für das letztendliche zu liefernde Ergebnis des Projekts verantwortlich, während die Product Owner der einzelnen Teams nur für jene Bestandteile und Features verantwortlich sind, die durch ihr jeweilige Scrum Teams entwickelt werden.

In einem großen Projekt ist es die Aufgabe des Chief Product Owners, die von den Product Ownern basierend auf ihrer Interaktion mit den Stakeholdern eingebrachten konkurrierenden Anforderungen zu priorisieren. Die Komplexität dieser Aufgabe nimmt mit der Zunahme der Anzahl von Scrum Teams und der Anzahl von Product Ownern drastisch zu. Ein wichtiges Element der Komplexität dieser Aufgabe ist es, sicherzustellen, dass die verschiedenen Bestandteile korrekt und zur richtigen Zeit integriert werden. Deshalb ist es unbedingt notwendig, eine Liste von Komponenten und Ressourcen zu entwickeln, die für alle Teams während des Projekts gemeinsam benötigt werden. Obwohl der Chief Product Owner die letztendlichen geschäftlichen Entscheidungen trifft, arbeitet er/sie mit dem Chief Scrum Master, anderen Product Ownern und Scrum Mastern zusammen, um diese Liste zu entwickeln.

Der Chief Product Owner spricht sich mit dem Programm Product Owner ab, um sicherzustellen, dass das große Projekt zu den Zielen des Programms kompatibel ist.

Chief Product Owner sollten sich sowohl die im SBOK® in den Leitfäden für Product Owner definierten Abschnitte als auch Kapitel 13, in dem die Skalierung von Scrum für große Projekte beschrieben wird, genauer ansehen.

3.7.2.2 Chief Scrum Master

Größere Projekte erfordern, dass mehrere Scrum Teams parallel arbeiten. Es mag nötig sein, gesammelte Informationen eines Teams in angemessener Weise an die anderen Teams weiterzugeben. Der Chief Scrum Master ist hierfür verantwortlich.

Die Rolle eines Chief Scrum Masters ist notwendig, um eine gute Zusammenarbeit zwischen den Scrum Teams sicherzustellen. Normalerweise erfolgt die Koordination mehrerer Scrum Teams eines Projekts durch das Scrum der Scrums (SoS) Meeting (siehe Abschnitt 13.2.2.1). Es gibt keine Hierarchie zwischen den Scrum Mastern: sie sind alle gleichberechtigt. Der Chief Scrum Master arbeitet ganz einfach auf einer Multiteamebene, während die Scrum Master auf der Ebene einzelner Teams arbeiten.

Abbildung 3-4 bietet Fragen die während eines Scrum der Scrums Meeting gestellt werden.

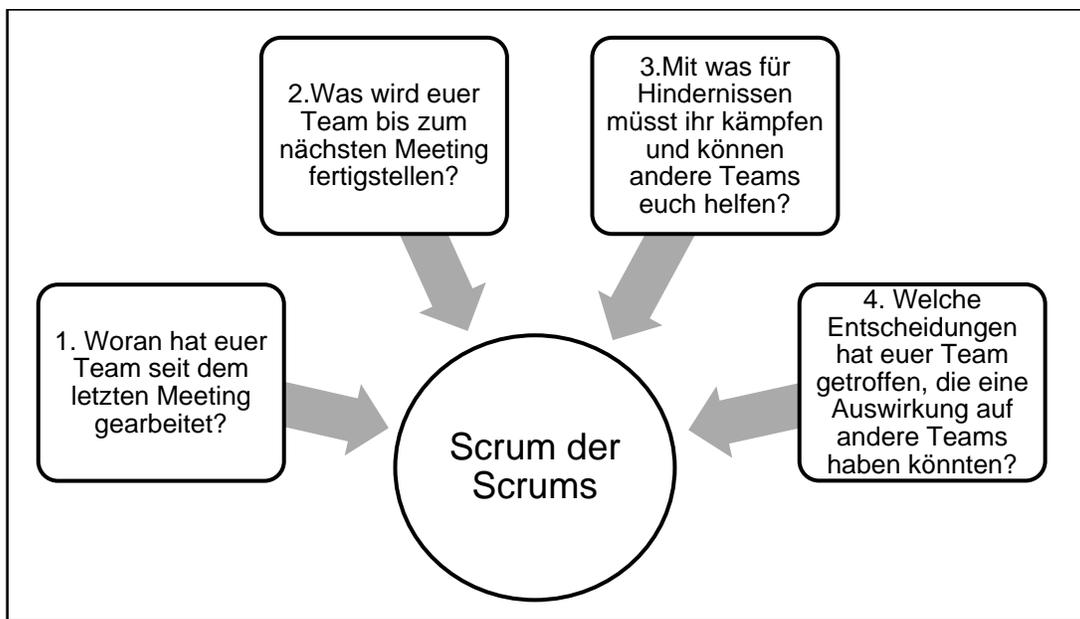


Abbildung 3-3: Gestellte Fragen während eines Scrum der Scrums Meeting

3.7.3 Anwendung von Scrum auf Unternehmens-Ebene

Die Anwendung von Scrum-Praktiken auf Projekte, die Teil eines Unternehmensmodells sind, erfordert ein Verständnis dafür, wie ein Unternehmen aufgebaut ist und gesteuert wird. Dies wird typischerweise mit Programmen und Portfolios gehandhabt.

Programm - Ein Programm ist eine Gruppe von zusammenhängenden Projekten mit dem Ziel, übergreifende Geschäftsergebnisse zu liefern, wie sie in der Programmvisionserklärung definiert sind. Das priorisierte Programm Backlog umfasst die priorisierten Produkt Backlogs für alle Projekte des Programms.

Portfolio - Ein Portfolio ist eine Gruppe von Programmen und/oder Projekten mit dem Ziel, die in der Portfoliovisionserklärung definierten Geschäftsergebnisse zu erzielen. Das priorisierte Portfolio-Backlog umfasst die priorisierten Programm-Backlogs für alle Programme im Portfolio und die priorisierten Produkt Backlogs der eigenständigen Projekte, die Teil des Portfolios sind.

Zu den Problemen und Fragen, die sich beim Einsatz von Scrum innerhalb eines Programms oder Portfolios stellen, gehört die Koordination zwischen zahlreichen Teams. Dies kann zum Scheitern führen, wenn es nicht sorgfältig gehandhabt wird. Die für die Kommunikation verwendeten Hilfsmittel müssen so skaliert werden, dass sie den Anforderungen der vielen an einem Programm oder Portfolio beteiligten Teams gerecht werden. Jedes Scrum-Team muss sich nicht nur um seine interne Kommunikation kümmern, sondern auch um die externe Kommunikation mit anderen Teams und den relevanten Business Stakeholdern des jeweiligen Programms oder Portfolios.

Bei der Anwendung von Scrum zum Management von Projekten im Rahmen eines Programms oder Portfolios wird dringend empfohlen, die in dieser Publikation vorgestellten allgemeinen Prinzipien von Scrum zu befolgen. Es versteht sich jedoch von selbst, dass geringfügige Anpassungen der Hilfsmittel und der Organisationsstruktur erforderlich sein können, um den Gesamtaktivitäten des Programms oder Portfolios und den gegenseitigen Abhängigkeiten Rechnung zu tragen. Wenn ein Scrum Guidance Body existiert, kann dieser dafür verantwortlich sein, die Organisation auf verschiedenen Ebenen zu überprüfen, um die angemessene Anwendung von Scrum zu verstehen und zu definieren, und als Beratungsgremium für alle, die an einem Projekt, Programm oder Portfolio arbeiten, zu fungieren.

Programme und Portfolios haben separate Teams mit unterschiedlichen Zielsetzungen. Ein Programm-Management-Team ist bestrebt, Fähigkeiten zu liefern und bestimmte Ziele zu verwirklichen, die zum Erreichen der spezifischen Programmziele beitragen. Dagegen muss ein Portfolio-Team die Ziele der verschiedenen zugehörigen Programme ausbalancieren, um die strategischen Ziele der Organisation als Ganzes zu erreichen.

Es ist wichtig zu beachten, dass bei der Skalierung von Scrum für das Unternehmen zusätzliche unterstützende Dienste wie Architekten, Produktmanager, Compliance, Informationssicherheit, Governance-Gremien usw. benötigt werden können.

Kapitel 14 enthält detaillierte Informationen zur Skalierung von Scrum für das Unternehmen.

3.7.4 Zusätzliche Kernrollen in Programmen und Portfolios

3.7.4.1 Programm Product Owner

Die Rolle des Programm Product Owners ähnelt der des Product Owners, mit dem Unterschied, dass sie darauf abzielt, die Bedürfnisse des Programms oder der Geschäftseinheit zu erfüllen und nicht die Bedürfnisse eines einzelnen Scrum-Teams.

Der Programm Product Owner legt die strategischen Ziele und Prioritäten eines Programms fest. Er oder sie ist für die Maximierung des Geschäftswerts des Programms verantwortlich, indem er oder sie die Kundenanforderungen klar artikuliert und die geschäftliche Rechtfertigung für das Programm aufrechterhält. Der Program Product Owner verwaltet auch das priorisierte Program Backlog. Er ist für die Erstellung und Verfeinerung der zu Liefernden Ergebnisse auf Programmebene verantwortlich und treibt diese voran, was die Koordination zwischen den zugrunde liegenden Projekten des Programms erfordert. Der Programm Product Owner ist auch für die Koordinierung mit anderen Programm Product Ownern verantwortlich, wenn andere Programme gemeinsame Abhängigkeiten und/oder gemeinsame Release-Pläne haben.

Der Program Product Owner stimmt sich auch mit dem zuständigen Portfolio Product Owner ab, um sicherzustellen, dass das Programm auf das entsprechende Portfolio abgestimmt ist. Der Program Product Owner arbeitet mit dem Portfolio Product Owner zusammen, um die Ausrichtung des Programms auf die Ziele des Portfolios sicherzustellen. Er oder sie ist auch an der Ernennung von Product Ownern für jedes einzelne zugehörige Projekt beteiligt und stellt sicher, dass die Vision, die Ziele, die Ergebnisse und die Releases der zugehörigen Projekte mit denen des Programms übereinstimmen.

Programm Product Owner sollten sich sowohl die im SBOK® in den Leitfäden für Product Owner definierten Abschnitte als auch Kapitel 14, in dem die Skalierung von Scrum für das Unternehmen beschrieben wird, genauer ansehen.

3.7.4.2 Portfolio Product Owner

Die Rolle des Portfolio Product Owners ähnelt der Rolle des Product Owners und auch der Rolle des Program Product Owners, mit dem Unterschied, dass sie darauf abzielt, die Bedürfnisse des Portfolios oder der Geschäftseinheit zu erfüllen und nicht die Bedürfnisse eines einzelnen Scrum-Teams oder die Bedürfnisse eines Programms.

Der Portfolio Product Owner trifft die Entscheidungen auf Portfolioebene. Er oder sie hat die beste Perspektive, um zu entscheiden, wie das Unternehmen organisiert werden soll, um die Vision zu erfüllen. Der Portfolio Product Owner ist für die Erstellung und Verfeinerung des Priorisierten Portfolio Backlogs verantwortlich und treibt diese voran

Portfolio Product Owner sollten sich sowohl die im SBOK® in den Leitfäden für Product Owner definierten Abschnitte als auch Kapitel 14, in dem die Skalierung von Scrum für das Unternehmen beschrieben wird, genauer ansehen.

3.7.4.3 Programm Scrum Master

Die Rolle des Program Scrum Masters ähnelt der des Scrum Masters, mit dem Unterschied, dass sie darauf abzielt, die Bedürfnisse des Programms oder der Geschäftseinheit zu erfüllen und nicht die Bedürfnisse eines einzelnen Scrum Teams.

Der Program Scrum Master ist ein Vermittler, der sicherstellt, dass alle Projektteams im Programm ein Umfeld vorfinden, das die erfolgreiche Durchführung ihrer Projekte ermöglicht. Der Programm Scrum Master leitet alle am Programm Beteiligten an, vermittelt und lehrt Scrum-Praktiken; er gibt den Scrum Mastern der einzelnen Projekte Anleitung; er räumt Hindernisse für die verschiedenen Projektteams aus dem Weg; er stimmt sich mit dem Scrum Guidance Body ab, um Ziele in Bezug auf Qualität, behördliche Vorschriften, Sicherheit und andere wichtige organisatorische Parameter zu definieren; und er stellt sicher, dass die Scrum-Prozesse im gesamten Programm effektiv eingehalten werden. Er oder sie ist ein Moderator, löst Probleme und beseitigt Hindernisse auf der Programmebene. Gleichzeitig ist er oder sie auch für die Koordination zwischen allen Projekten im Programm und für die Koordination mit anderen Programmen mit gemeinsamen Abhängigkeiten oder gemeinsamen Releaseplänen verantwortlich.

Der Programm Scrum Master arbeitet mit dem Portfolio Scrum Master zusammen, um die Ausrichtung des Programms an den Zielen des Portfolios sicherzustellen. Er oder sie ist auch an der Ernennung von Scrum Mastern für einzelne Projekte beteiligt und stellt sicher, dass die Vision, die Ziele, die Ergebnisse und die Releases jedes Projekts im Programm mit denen des Programms übereinstimmen.

Programm Scrum Master sollten sich sowohl die im SBOK® in den Leitfäden für Scrum Master definierten Abschnitte als auch Kapitel 14, in dem die Skalierung von Scrum für das Unternehmen beschrieben wird, genauer ansehen.

3.7.4.4 Portfolio Scrum Master

Die Rolle des Portfolio Scrum Masters ähnelt der Rolle des Scrum Masters, mit dem Unterschied, dass sie darauf abzielt, die Bedürfnisse des Portfolios oder der Geschäftseinheit zu erfüllen und nicht die Bedürfnisse eines einzelnen Scrum Teams.

Portfolio Scrum Master sollten sich sowohl die im SBOK® in den Leitfäden für Scrum Master definierten Abschnitte als auch Kapitel 14, in dem die Skalierung von Scrum für das Unternehmen beschrieben wird, genauer ansehen.

3.7.5 Beispiele von Projekten, Programmen und Portfolios

Es folgen Beispiele von Projekten, Programmen und Portfolios aus verschiedenen Branchen und Bereichen:

Beispiel 1: Baufirma

- Projekt—Bau eines Hauses
- Programm—Bau einer Wohnsiedlung
- Portfolio—Alle Wohnprojekte der Firma

Beispiel 2: Luft- und Raumfahrtgesellschaft

- Projekt—Bau einer Trägerrakete
- Programm—Erfolgreicher Start eines Satelliten
- Portfolio—Alle aktiven Satelliten Programme

Beispiel 3: Informationstechnologie (IT) Unternehmen

- Projekt—Entwicklung des Einkaufswagen Moduls
- Programm—Erstellung einer voll funktionierenden E-Commerce Website
- Portfolio—Alle bisher von dem Unternehmen erstellten Websites

Abbildung 3-4 zeigt, wie Scrum in einer Organisation für Portfolios, Programme oder Projekte angewendet werden kann.

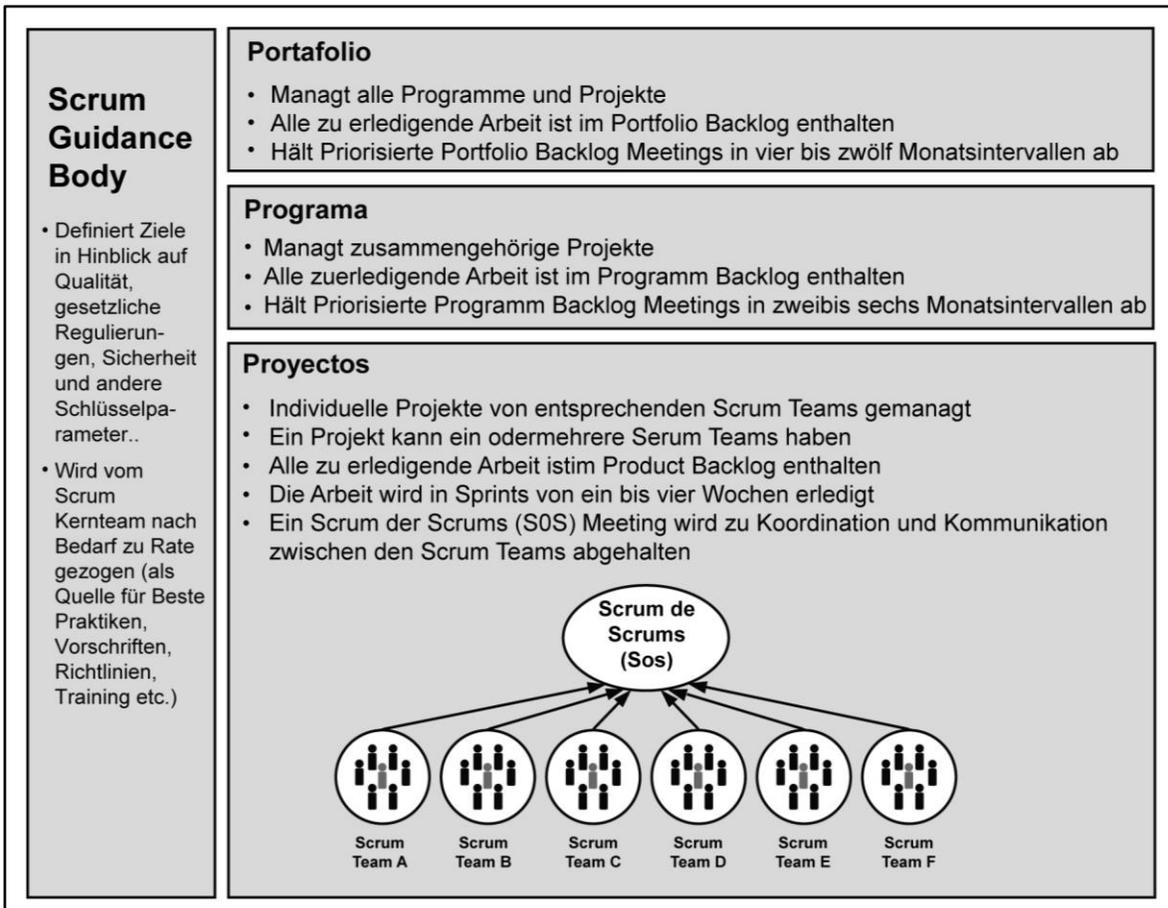


Abbildung 3-4: Scrum quer durch die Organisation für Projekte, Programme und Portfolios

3.8 Zusammenfassung der Verantwortlichkeiten

Rollen	Verantwortlichkeiten
Scrum Guidance Body	<ul style="list-style-type: none"> • Stellt allgemeine Richtlinien und Metriken zur Erstellung der Rollen beschreibungen von Scrum Teammitgliedern • Wirkt als ein Berater für Projekte verschiedener Ebenen in der gesamten Organisation • Versteht und definiert angemessene Level der Gruppierung, Rollen und Meetings in Scrum Projekten
Portfolio Product Owner	<ul style="list-style-type: none"> • Definiert die strategischen Ziele und Prioritäten der Portfolios
Portfolio Scrum Master	<ul style="list-style-type: none"> • Löst Probleme und koordiniert Meetings für Portfolios
Programm Product Owner	<ul style="list-style-type: none"> • Definiert die strategischen Ziele und Prioritäten der Programme
Programm Scrum Master	<ul style="list-style-type: none"> • Löst Probleme und koordiniert Meetings für Programme
Business Stakeholdern	<ul style="list-style-type: none"> • Schließt als umfassender Begriff Kunden, Anwender und Sponsoren ein • Interagiert regelmäßig mit Product Owner, Scrum Master und Scrum Team, um Input zu liefern und unterstützt die Schaffung der mit dem Projekt verbundene zu liefernden Ergebnisse
Product Owner/Chief Product Owner	<ul style="list-style-type: none"> • Erstellt die anfänglichen allgemeinen Anforderungen des Projekts und bringt das Projekt auf den Weg • Ernennet geeignete Personen zu Scrum Master und Scrum Teammitgliedern • Bietet die anfänglichen und laufenden finanziellen Ressourcen für das Projekt • Bestimmt die Produktvision • Schätzt die Durchführbarkeit ein und stellt die Lieferung des Produkts oder der Dienstleistung sicher • Gewährleistet Transparenz und Übersichtlichkeit der Priorisierten Produkt Backlog Einträge • Bestimmt Minimalen vermarktbar Release-Inhalt • Liefert Abnahmekriterien für die in einem Sprint zu erstellenden User Storys • Überprüft zu liefernde Ergebnisse • Bestimmt die Sprint-Länge
Scrum Master/Chief Scrum Master	<ul style="list-style-type: none"> • Stellt sicher, dass alle Teammitglieder, einschließlich des Product Owners, die Scrum Prozesse korrekt befolgen • Stellt ein reibungsloses Fortschreiten der Entwicklung des Produkts oder der Dienstleistung sicher, und dass die Scrum Teammitglieder alle benötigten Hilfsmittel haben, die Arbeit zu vollenden • Beaufsichtigt das Release Plan Meeting und plant andere Meetings
Scrum Team	<ul style="list-style-type: none"> • Übernimmt kollektive Verantwortung und stellt sicher, dass die zu liefernden Ergebnisse des Projekts den Anforderungen entsprechend entwickelt werden • Versichert dem Product Owner und Scrum Master, dass die zugeteilte Arbeit nach Plan verrichtet wird

Tabelle 3-4: Zusammenfassung der Verantwortlichkeiten in Bezug zur Organisation

3.9 Scrum versus traditionelles Projekt Management

Organisationsstruktur und Definition der Rollen und zugehörigen Verantwortlichkeiten sind einige der Bereiche, in denen Scrum sich wesentlich von traditionellen Projekt Management Methoden unterscheidet.

In traditionellen Projekt Management Methoden ist die Organisationsstruktur hierarchisch und die Befugnis für alle Aspekte des Projekts wird von der höheren Ebene zur niedrigen Ebene delegiert (z.B. Projekt Sponsor delegiert Handlungsgewalt an den Projekt Manager und der Projekt Manager delegiert diese an Team Manager oder -Mitglieder). Traditionelle Projekt Management Methoden unterstreichen die individuelle Rechenschaft für Projekt Verantwortlichkeiten anstatt Gruppenverantwortung oder -rechenschaft. Jegliche Abweichung von der delegierten Handlungsbefugnis wird als Problem angesehen und kann an eine höhere Ebene der Organisationshierarchie weitergegeben werden. In der Regel ist der Projekt Manager für den erfolgreichen Abschluss des Projekts verantwortlich und er/sie trifft die Entscheidungen für verschiedene Aspekte des Projekts, einschließlich Initiierung, Planung, Beurteilung, Ausführung, Überwachung, Kontrolle und Abschluss.

Scrum legt die Betonung auf Selbstorganisation und Selbstmotivierung, wobei das Team größere Verantwortung übernimmt, um das Projekt zum Erfolg zu führen. Dies stellt sicher, dass das Team hinter dem Projekt steht und gemeinsame Verantwortung entsteht. Dies seinerseits läuft auf Motivation des Teams hinaus, was zur Optimierung der Team Effizienz führt. Product Owner, Scrum Master und das Scrum Team arbeiten eng mit relevanten Business Stakeholdern an der Verfeinerung der Anforderungen während der *Epics entwickeln*, *Priorisiertes Produkt Backlog erstellen* und *User Storys Erstellen* Prozesse zusammen. Dies gewährleistet, dass es keinen Spielraum für isoliertes Planen in Scrum gibt. Team Erfahrung und Fachwissen werden in der Produkt Entwicklung benutzt, um die benötigten Inputs zur Planung, Abschätzung und Umsetzung der Projektarbeit zu bewerten. Und die Zusammenarbeit der Scrum Kernteammitglieder garantiert ein innovatives und kreatives Umfeld zur Durchführung des Projekts, was das Wachstum fördert und zur Teamharmonie beiträgt.

3.10 Populäre Theorien aus dem Personalbereich und deren Relevanz in Scrum

3.10.1 Modell der Gruppendynamik nach Tuckman

Scrum Ansatz und Methode mögen anfangs sehr fremd und schwierig für ein neues Scrum Team aussehen. Ein neues Scrum Team, wie jedes andere neue Team auch, entwickelt sich generell durch einen vier-Phasen-Prozess während des ersten Scrum Projekts. Dieser Prozess ist als Modell der Gruppendynamik nach Tuckman (Tuckman, 1965) bekannt. Die Grundidee dahinter ist, dass die vier Phasen—Forming (Einstiegs- und Findungsphase), Storming (Streit- und Auseinandersetzungsphase), Norming (Regel- und Übereinstimmungsphase) und Performing (Arbeits- und Leistungsphase)— für ein Team notwendig sind, um sich durch Mindern von Problem und Herausforderungen zu entwickeln, Lösungen zu finden, Arbeit zu planen und Ergebnisse zu liefern. Ein Scrum Master sollte sich der Phase bewusst sein, in der sich das Scrum-Team befindet, und dem Team helfen, ein leistungsfähigeres Team zu werden.

Die vier Phasen des Modells lauten wie folgt:

1. **Forming**—Dies wird oft als eine freudige Phase empfunden, denn alles fühlt sich neu an und das Team ist noch auf keine Schwierigkeiten im Projekt gestoßen.
2. **Storming**—In dieser Phase versucht das Team seine Arbeit zu verrichten; allerdings können Machtkämpfe aufkommen und es kommt häufig zu Chaos und Verwirrung innerhalb des Teams.
3. **Norming**—Nun beginnt das Team zu reifen, bereinigt interne Unstimmigkeiten und findet Lösungen durch Zusammenarbeit. Diese Phase wird als Phase der Anpassung angesehen.
4. **Performing**—Während dieser Phase wird das Team am meisten zusammengeschweißt und wirkt auf dem höchsten Leistungsniveau. Die Mitglieder haben sich zu einem effizienten Team von professionellen Fachkollegen entwickelt, die beständig produktiv sind.

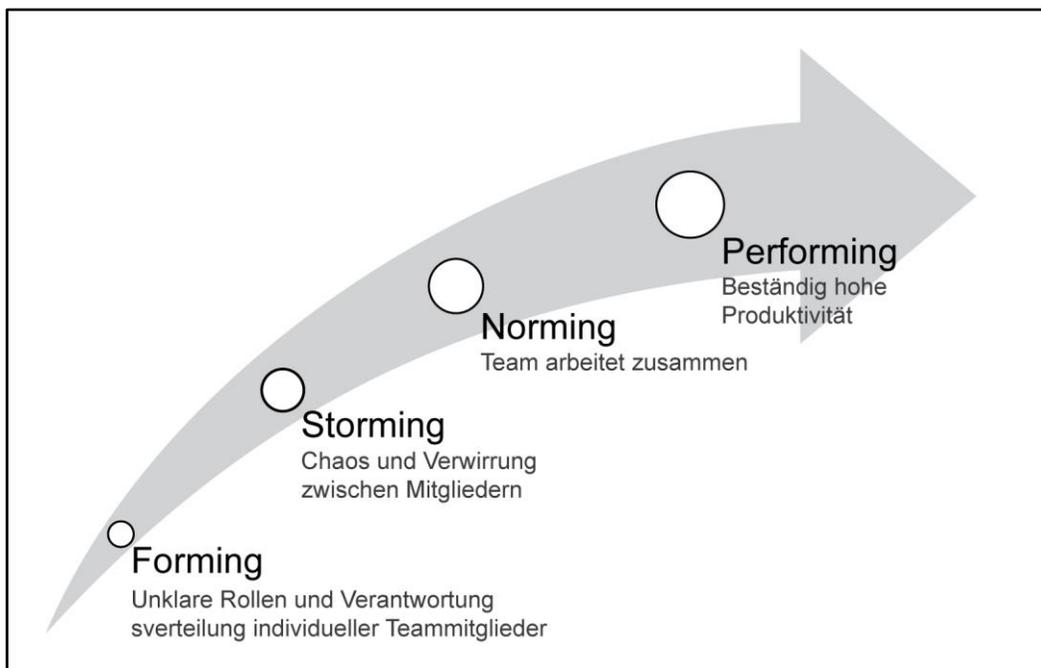


Abbildung 3-5: Phasen der Gruppenentwicklung nach Tuckman

3.10.2 Konfliktmanagement

Organisationen die den Scrum Rahmen anwenden, fördern ein offenes Umfeld und Dialoge zwischen Mitarbeitern. Konflikte zwischen Scrum Teammitgliedern werden im Allgemeinen eigenständig, mit wenig oder ohne jegliche Einbeziehung des Managements oder anderer außerhalb des Scrum Teams gelöst.

Konflikte können zuträglich sein, wenn sie Diskussionen vorantreiben und Debatten anregen, denn zumeist hat dies einen Nutzen für das Projekt und die betreffenden Teammitglieder zur Folge. Es ist daher wichtig, dass zur Lösung von Konflikten ermutigt wird, ein offenes Klima gefördert wird, in dem Teammitglieder sich frei fühlen, ihre Meinungen und Bedenken voneinander und über das Projekt zu äußern und schlussendlich übereinstimmen, was zu leisten ist und wie die Arbeit eines jeden Sprints ausgeführt wird.

Konflikt Management Techniken werden von Team Mitgliedern zur Lösung von Konflikten verwendet, die während eines Scrum Projekts entstehen. Konfliktquellen sind oftmals Zeitpläne, Prioritäten, Ressourcen, Hierarchie, technische Probleme, Vorgehensweisen, Persönlichkeiten und Kosten.

3.10.3 Konfliktmanagement Techniken

Gewöhnlich gibt es in einer Organisation, die Scrum Prozesse anwendet, vier Vorgehensweisen um Konflikte anzugehen:

1. Win/Win
2. Lose/Win
3. Lose/Lose
4. Win/Lose

3.10.3.1 Win/Win

Es ist normalerweise am besten für Teammitglieder, die Probleme direkt mit einer kooperativen Einstellung und einem offenen Dialog anzugehen, um jegliche Unstimmigkeiten auszuräumen und einen Konsens zu erreichen. Dieser Ansatz wird als *Win-Win* bezeichnet. Organisationen die Scrum anwenden sollten ein Umfeld fördern, in dem Mitarbeiter sich frei fühlen, offen Problemen oder Angelegenheiten ins Auge zu sehen, sie zu besprechen und zu lösen, um ein *Win/Win* Resultat zu erzielen.

3.10.3.2 Lose/Win

Manche Teammitglieder mögen sich zeitweise fühlen, als ob ihre Beiträge von anderen nicht anerkannt oder wertgeschätzt werden, oder dass sie nicht gleich behandelt werden. Dies kann dazu führen, dass sie aufhören, einen effektiven Beitrag für das Projekt zu leisten und allem was ihnen gesagt wird zustimmen, auch wenn sie anderer Meinung sind.

Dieser Ansatz wird als *Lose/Win* bezeichnet. Diese Situation mag auftreten, wenn es Mitglieder im Team gibt (Manager eingeschlossen), die eine autoritäre oder forschende Art haben, Anweisungen zu geben und/oder nicht alle Teammitglieder gleich behandeln. Dieser Ansatz ist keine gewünschte Konfliktmanagement Technik für Scrum Projekte, da aktive Beiträge von jedem Mitglied des Teams für die erfolgreiche Beendung eines jeden Sprints zwingend notwendig sind. Der Scrum Master sollte die Beteiligung jedes Teammitgliedes anregen, das sich von Konfliktsituationen zurückziehen scheint.

Zum Beispiel ist es für alle Teammitglieder in den Täglichen Standup Meetings wichtig zu sprechen und dazu beizutragen, dass alle Angelegenheiten oder Hindernisse bekannt und effektiv behandelt werden.

3.10.3.3 Lose/Lose

In Konfliktsituationen mag es vorkommen, dass ein Teammitglied versucht zu verhandeln oder nach Lösungen sucht, die nur eine teilweise oder kurzfristige Zufriedenheit der streitenden Parteien zur Folge hätten. Dies könnte in Scrum Teams passieren, wenn Teammitglieder versuchen eine suboptimale Lösung für ein Problem zu verhandeln. Dieser Ansatz beinhaltet typischerweise ein „Geben und Nehmen“, um jedes Teammitglied zufrieden zu stellen—anstatt zu versuchen, das wirkliche Problem zu lösen. In der Regel endet dies in einem allgemeinen *Lose-Lose* Ergebnis für alle Beteiligten und demzufolge für das Projekt. Das Scrum Team sollte sorgsam darauf bedacht sein, dass Teammitglieder nicht in eine *Lose/Lose* Mentalität fallen. Tägliche Standups und andere Scrum Meetings werden abgehalten, um sicherzustellen, dass derzeitige Probleme durch gemeinsame Besprechungen gelöst werden.

3.10.3.4 Win/Lose

Zeitweise mag der Scrum Master oder ein anderes einflussreiches Teammitglied denken, dass er/sie Anführer oder Manager des Teams sei und eigene Ansichten auf Kosten der anderen durchsetzen muss. Diese Konfliktmanagement Technik wird oft durch Konkurrenzdenken gekennzeichnet und endet normalerweise mit einem *Win-Lose* Ergebnis. Dieser Ansatz wird für Scrum Projekte nicht empfohlen, da Scrum Teams von Natur aus selbstorganisiert und ermächtigt sind, in denen keine Einzelperson Autorität über andere Teammitglieder hat. Auch wenn das Scrum Team Personen mit verschiedenen Niveaus an Erfahrungen und Fachwissen enthalten kann, wird jedes Mitglied gleichbehandelt und niemand hat das Recht, der alleinige Entscheidungsträger zu sein.

3.10.4 Führungsstile

Führungsstile variieren je nach Organisation, Situation und sind sogar von den Individuen und den Zielen des Scrum Projekts abhängig. Einige übliche Führungsstile sind folgende:

- **Unterstützender Leader Stil**—Unterstützende Leader benutzen Zuhören, Empathie, Engagement und Einsicht, während sie Macht und Autorität mit den Teammitgliedern teilen. Dienende Leader sind Stewards, die Ergebnisse erzielen, indem sie sich auf die Bedürfnisse des Teams konzentrieren. Dieser Stil ist die Verkörperung der Scrum Master Rolle.
- **Delegieren**—Delegierende Leader sind in einen Großteil der Entscheidungsfindung eingebunden; jedoch geben sie einen Teil der Planungs- und Entscheidungsverantwortung an Teammitglieder weiter, insbesondere, wenn diese fähig sind, Aufgaben kompetent umzusetzen. Dieser Führungsstil eignet sich für Situationen, in denen der Leader mit speziellen Projektdetails konform geht und Zeit begrenzt ist.

- **Autokratisch**—Autokratische Leader treffen Entscheidungen auf eigene Faust und erlauben somit Teammitgliedern wenig bis keine Teilnahme an oder Auseinandersetzung mit der Entscheidungsfindung. Dieser Führungsstil sollte nur selten verwendet werden.
- **Anweisend**—Anweisende Leader weisen ihre Teammitglieder an, welche Aufgaben zu leisten sind sowie auf welche Art und zu welchem Zeitpunkt diese auszuführen sind.
- **Laissez Faire**—Ein Führungsstil, der das Team weitestgehend unbeaufsichtigt und ohne Eingriff des Leaders in den Arbeitsalltag belässt. Dieser Stil führt häufig zu einem anarchischen Zustand.
- **Coachen/Unterstützen**—Coachende und Unterstützende Leader geben Anweisungen und unterstützen und beaufsichtigen dann die Teammitglieder durch Zuhören, Hilfestellungen, Ermuntern und eine positive Auffassung in Zeiten der Unsicherheit.
- **Aufgabenorientiert**—Aufgabenorientierte Leader setzen Aufgabenerledigung und Termintreue durch.
- **Selbstbewusst**—Selbstbewusste Leader stellen sich Problemen und treten selbstsicher auf, um sich mit Respekt Autorität zu verschaffen.

3.10.4.1 Unterstützender Leader Stil

Der bevorzugte Führungsstil für ein Scrum Projekt ist der Unterstützender Leader Stil. Larry Spears identifiziert zehn Merkmale, die jeder effektive Dienende Leader haben sollte:

1. **Zuhören**—Von Leadern wird erwartet, dass sie aufmerksam und aufnahmebereit zuhören was gesagt oder nicht gesagt wird. Sie sind in der Lage, auf ihre innere Stimme zu hören, um ihre eigenen Gefühle zu verstehen und darüber zu sinnieren.
2. **Empathie**—Gute Leader akzeptieren und anerkennen Individuen für ihre speziellen und einzigartigen Fähigkeiten und Talente. Sie gehen davon aus, dass Mitarbeiter gute Absichten haben und akzeptieren sie als Individuen, auch wenn sie Verhaltens- oder Leistungsprobleme haben.
3. **Heilung**—Die Motivation und das Potenzial, sich selbst und seine Beziehungen mit anderen zu heilen, ist ein starkes Merkmal des Dienenden Leaders. Dienende Leader erkennen, wenn ihre Kollegen emotional leiden und ergreifen die Initiative um ihnen zu helfen.
4. **Wahrnehmung**—Wahrnehmung und speziell Selbstwahrnehmung ist ein Merkmal eines Leaders. Dies ermöglicht ihm, Angelegenheiten in Bezug auf Ethik, Macht und Werte besser zu verstehen und zu integrieren.
5. **Überzeugung**—Leader gebrauchen Überzeugung und nicht ihre Autoritätsposition, um einen Gruppenkonsens zu erreichen und Entscheidungen zu fällen. Dienende Leader üben sich in Überzeugung, anstatt Zustimmung zu erzwingen, was in manchen autoritären Managementstilen üblich ist.

6. **Konzeptualisierung**—Die Fähigkeit, Probleme (oder eine Organisation) aus einer breiteren, konzeptionellen, visionären Perspektive zu sehen und zu analysieren, anstatt nur auf die unmittelbaren, kurzfristigen Ziele zu fokussieren, ist eine einzigartige Fähigkeit eines guten Leaders.
7. **Voraussicht**—Sein intuitiver Verstand ermöglicht Leadern, früher Gelerntes und gegenwärtige Gegebenheiten zu benutzen und anzuwenden, um das Ergebnis aktueller Situationen und Entscheidungen abzusehen.
8. **Verwaltung**—Die Verwaltung fordert Hingabe, anderen zu dienen. Leader ziehen Überzeugung der Kontrolle vor um sicherzustellen, dass sie das Vertrauen anderer in der Organisation gewinnen.
9. **Verpflichtung zum Wachstum anderer**—Leader haben eine starke Verpflichtung zum Wachstum anderer Leute in ihrer Organisation. Sie übernehmen die Verantwortung, die persönliche, professionelle und geistige Weiterentwicklung anderer zu fördern (z.B. Zugang zu Ressourcen für persönliche oder professionelle Entwicklung zu bieten, Mitarbeiter zur Teilnahme an Entscheidungsprozessen zu animieren).
10. **Aufbauen von Gemeinschaften**—Leader sind daran interessiert, Gemeinschaften innerhalb des Arbeitsumfelds aufzubauen, besonders im Angesicht der Veränderung der Gesellschaft, weg von kleineren Gemeinschaften hin zu großen Institutionen, die das Leben der Menschen formt und kontrolliert.

Scrum glaubt, dass alle Leader in Scrum Projekten (Scrum Master und Product Owner eingeschlossen) Unterstützende Leader mit den oben genannten Merkmalen sein sollten.

3.10.5 Hierarchie der Bedürfnisse Theorie nach Maslow

Maslow (1943) stellt eine Bedürfnis Hierarchie vor, welche anerkennt, dass verschiedene Menschen auf verschiedenen Bedürfnisebenen sind. In der Regel fangen Menschen an nach physiologischen Bedürfnissen zu schauen und gehen dann stufenweise die Bedürfnis Hierarchie nach Oben.

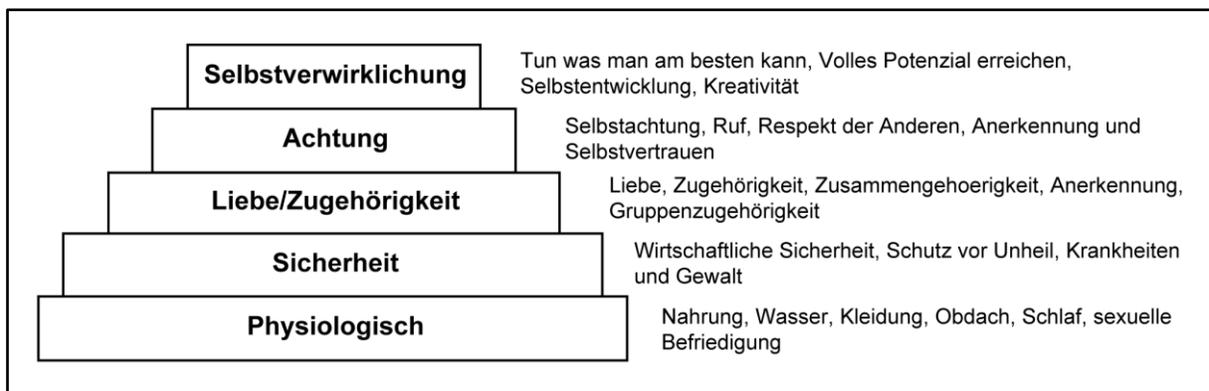


Abbildung 3-6: Hierarchie der Bedürfnisse Theorie nach Maslow

Um erfolgreich zu sein benötigt das Scrum Team sowohl Kernteammitglieder, als auch Nicht-zentrale Teammitglieder, welche den Achtungs- oder Selbstverwirklichungs-Level erreicht haben. Das Konzept der selbstorganisierten Teams, welches ein Schlüsselprinzip in Scrum ist, benötigt selbst-motivierte Teammitglieder, welche volle Anteilnahme und einen vollen Beitrag dazu leisten, die Projektziele zu erreichen.

Als ein Leader muss der Scrum Master verstehen, wo jede Person des Teams in Relation zur Pyramide steht. Dieses Verständnis wird helfen, den besten Ansatz zur Motivation jedes Einzelnen zu finden.

Hinzukommt, dass jeder im Laufe seines Lebens die Ebenen der Bedürfnis Hierarchie hoch und runter steigt, aufgrund seiner eigenen Motivation und Anstrengungen aufsteigt oder manchmal aufgrund von Faktoren außerhalb seiner Kontrolle hinunter gestoßen wird. Das Ziel des Scrum Masters ist, mit Individuen seines Teams zu arbeiten, um deren Fähigkeiten und Wissen auszubauen und ihnen zu helfen, in der Bedürfnishierarchie aufzusteigen. Diese Unterstützung führt zu einem Team von Individuen, die motiviert sind und einen großen Beitrag zum Projekt und zur Organisation als Ganzem leisten.

3.10.6 X-Theorie und Y-Theorie

Douglas McGregor (1960) stellte zwei Management Theorien auf:

- **X-Theorie** — X-Theorie Leader sind der Annahme, dass Mitarbeiter von Natur aus unmotiviert sind und Arbeit wenn möglich vermeiden, wodurch ein autoritärer Führungsstil gerechtfertigt wird.
- **Y-Theorie** — Y-Theorie Leader sind der Annahme, dass Mitarbeiter Eigenmotivation besitzen und nach größerer Verantwortung streben, weshalb die Y-Theorie mit einem partizipativeren Führungsstil einhergeht.

Abraham H. Maslow (1960) schlug die Theorie Z vor, und William Ouchi (1980) lieferte eine weitere Version der Theorie Z, die die Theorie X und die Theorie Y erweitert:

- **Theorie Z** - In Maslows Version gehen Theorie Z Leader davon aus, dass Mitarbeiter motiviert werden können, indem sie ihr Streben nach Selbsttranszendenz nutzen, ohne ihre Motivationen im Zusammenhang mit der Hierarchie der Bedürfnisse zu ignorieren. In der Version von Ouchi gehen Führungskräfte der Theorie Z davon aus, dass Mitarbeiter motiviert werden können, indem Stabilität durch Arbeitsplatzsicherheit, hohe Arbeitsmoral und Zufriedenheit sowohl am Arbeitsplatz als auch außerhalb des Arbeitsplatzes gefördert wird.

In Organisationen mit X-Theorie Leadern in den Rollen des Scrum Master oder Product Owners haben Scrum Projekte wenig Chancen auf Erfolg. Alle Leader in Scrum Projekten sollten die Y-Theorie übernehmen, wonach sie Einzelpersonen als wichtiges Kapital ansehen und darauf hinarbeiten, die Fähigkeiten ihrer Teammitglieder auszubauen und durch ausdrückliche Anerkennung ihrer Arbeit ihre Teammitglieder zum Erreichen der Projektziele zu befähigen.

4. GESCHÄFTLICHE RECHTFERTIGUNG

4.1 Einleitung

Das Ziel dieses Kapitels ist es, das Konzept und den Zweck der Geschäftlichen Rechtfertigung in Bezug auf Scrum Projekte zu verstehen. Es ist für eine Organisation wichtig, vor dem Start eines Projekts eine geeignete Geschäftliche Rechtfertigung durchzuführen und eine realisierbare Projektvisionserklärung zu erstellen. Dies hilft den Hauptentscheidungsträgern, den Geschäftsbedarf nach Veränderung und für ein neues Produkt oder eine Dienstleistung zu verstehen und gibt die Rechtfertigung, mit einem Projekt voranzuschreiten. Außerdem hilft es dem Product Owner, ein Priorisiertes Produkt Backlog zusammen mit den Geschäftserwartungen des Senior Managements & der Business Stakeholdern zu erstellen.

Geschäftliche Rechtfertigung, wie im *Scrum Body of Knowledge (SBOK®)* definiert, ist auf Folgendes anwendbar:

- Portfolios, Programme und/oder Projekte in jeder Branche
- Produkte, Dienstleistungen oder jegliche an Business Stakeholdern zu liefernde Ergebnisse
- Projekte jeglicher Größe oder Komplexität

Der Begriff "Produkt" in dieser Wissenssammlung kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann auf jedes Projekt in jeder Branche effektiv angewendet werden — von kleinen Projekten oder Teams mit lediglich sechs Team-Mitgliedern bis hin zu großen, komplexen Projekten mit mehreren Hundert Team-Mitgliedern.

Dieses Kapitel ist in folgende Abschnitte aufgeteilt:

4.2 Leitfaden für die Kernrollen—Dieser Abschnitt klärt die Frage, welche Abschnitte für jede Scrum Rolle relevant sind: Product Owner, Scrum Master und Scrum Team.

4.3 Wertgetriebene Lieferung—Dieser Abschnitt beschreibt das Konzept des Geschäftswertes und seine Bedeutung in jedem Projekt. Außerdem bietet er Informationen bezüglich der Verantwortlichkeiten der verschiedenen am Erreichen des Geschäftswertes beteiligten Einzelpersonen, einschließlich des Product Owners.

4.4 Wichtigkeit der Geschäftlichen Rechtfertigung—Dieser Abschnitt geht näher auf die Wichtigkeit der Geschäftlichen Rechtfertigung ein, welche Faktoren sie bestimmen und wie sie durch das Projekt hindurch erhalten und nachgewiesen wird.

4.5 Prozess der Geschäftlichen Rechtfertigung—Dieser Abschnitt beschreibt im Detail, wie die Geschäftliche Rechtfertigung unter Benutzung verschiedener Hilfsmittel berechnet und nachgewiesen wird und erklärt dann, wie Nutzen realisiert und bestätigt wird.

4.6 Kontinuierliche Wertrechtfertigung— Dieser Abschnitt beschreibt die Bedeutung von Kontinuierlicher Wertrechtfertigung und erklärt wie sie erzielt wird.

4.7 Bestätigen der Ertragsrealisierung— Dieser Abschnitt beschreibt wie Nutzen und Ertrag im ganzen Projekt realisiert werden.

4.8 Zusammenfassung der Verantwortlichkeiten—Dieser Abschnitt definiert die Verantwortlichkeiten der ProjektTeammitglieder in Bezug auf die Geschäftliche Rechtfertigung, basierend auf ihren Rollen.

4.9 Scrum versus traditionelles Projekt Management—Dieser Abschnitt hebt die Geschäftsvorteile von Scrum gegenüber den traditionellen Projekt Management Modellen hervor.

4.2 Leitfaden für die Kernrollen

1. Product Owner —Die Geschäftliche Rechtfertigung wird hauptsächlich vom Product Owner durchgeführt; demnach ist das gesamte Kapitel insbesondere für diese Kernrolle relevant.
2. Scrum Master—Der Scrum Master sollte mit diesem kompletten Kapitel vertraut sein, besonders mit den Abschnitten 4.3, 4.4, 4.6, 4.7, und 4.8.
3. Scrum Team—Das Scrum Team sollte sich hauptsächlich auf die Abschnitte 4.3, 4.7, und 4.8 konzentrieren.

4.3 Wert-getriebene Lieferung

Unter einem Projekt versteht man ein gemeinschaftliches Unternehmen zur Erstellung von neuen Produkten oder Dienstleistungen oder Ergebnissen, wie sie in der Projektvisionserklärung definiert sind. Einschränkungen, denen Projekte in der Regel unterliegen, sind Zeit, Kosten, Lieferumfang, Qualität, Personen und die Fähigkeiten der Organisation(en). In der Regel wird von den erzeugten Ergebnissen eines Projekts erwartet, in der einen oder anderen Weise einen Geschäftsmehrwert oder Dienstleistungsmehrwert zu schaffen.

Da der primäre Grund mit einem Projekt zu starten für jede Organisation der Mehrwert ist, muss der Schwerpunkt auf der wertgetriebenen Lieferung liegen. Werte zu liefern ist tief im Scrum Rahmen verwurzelt. Scrum unterstützt die frühe Lieferung von Werten und fährt damit den ganzen Projektzyklus hindurch fort.

Einer der Hauptmerkmale eines jeden Projekts ist die Ungewissheit der Ergebnisse oder des Ausgangs. Unabhängig von der Größe oder Komplexität eines Projekts es ist unmöglich, eine Garantie für den Erfolg eines Projekts zu geben. In Anbetracht diese Unsicherheit auf Erfolg ist es daher wichtig, so früh wie möglich im Projekt damit anzufangen, Ergebnisse zu liefern. Solch frühe Lieferung von Ergebnissen und dementsprechenden Werten bietet eine Möglichkeit zur Reinvestition und belegt den Wert eines Projekts für interessierte Business Stakeholdern.

Für wertgetriebene Lieferung ist es wichtig:

1. Zu verstehen, was dem Kunden und Anwender einen Mehrwert gibt und den Anforderungen von hohem Wert im Priorisierten Produkt Backlog den Vorrang zu geben.
2. Ungewissheiten zu verringern und Risiken fortlaufend zu adressieren, die (falls sie auftauchen) potenziell den Wert vermindern können. Es ist auch wichtig, eng mit Projekt Business Stakeholdern zusammenzuarbeiten, die erstellten Produktinkremente vorzuführen, sowie ihnen ein effektives Änderungsmanagement zu ermöglichen.
3. *Zu liefernde Ergebnisse zu erzeugen*, basierend auf den festgelegten Prioritäten, indem potenziell versandfähige zu liefernde Produktinkremente während jedes Sprints erstellt werden, wodurch der Kunde früh im Projekt einen Mehrwert erzielen kann.

Das Konzept der wertgetriebenen Lieferung in Scrum macht den Scrum Rahmen sehr attraktiv für geschäftliche Business Stakeholdern und das Senior Management. Dieses Konzept unterscheidet sich sehr von den traditionellen Projekt Management Modellen, in denen:

1. Anforderungen nicht nach Geschäftswert priorisiert werden.
2. es schwierig ist, Anforderungen nach Projektbeginn zu ändern, da dies nur durch einen zeitaufwendigen Änderungsmanagement Prozess möglich ist.
3. ein Mehrwert erst am Ende des Projekts mit der Lieferung des Endprodukts oder der Dienstleistung erzielt wird.

Abbildung 4-1 vergleicht die wertgetriebene Lieferung in Scrum versus traditionellen Projekten.

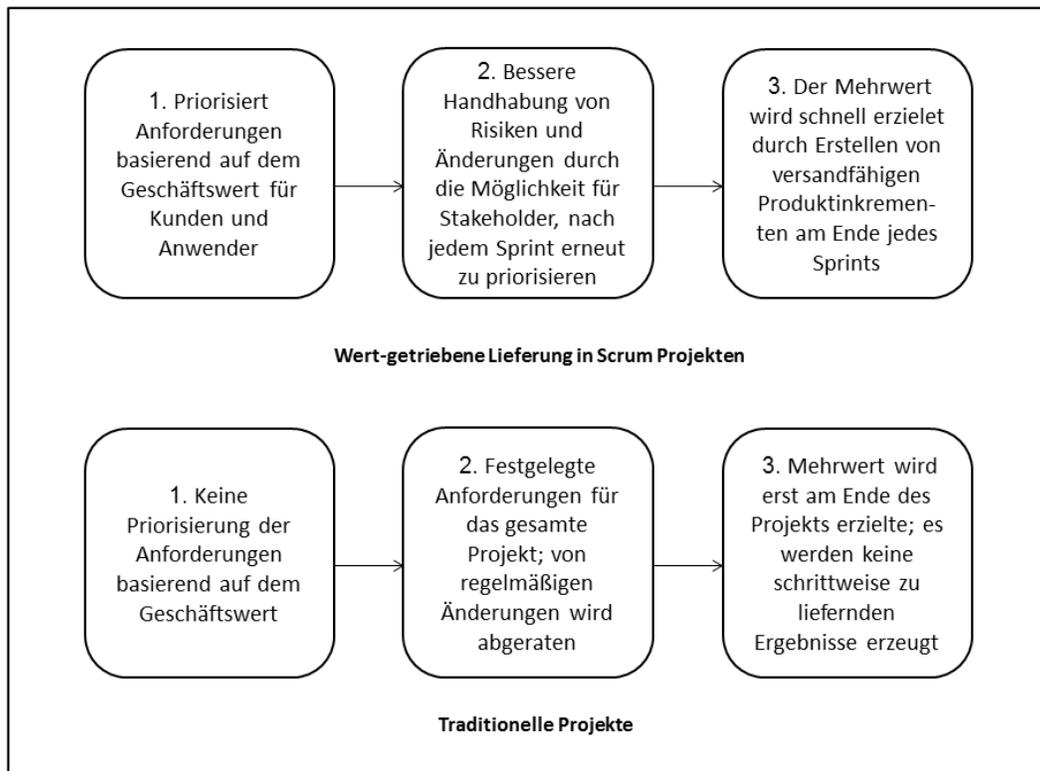


Abbildung 4-1: Lieferung von Mehrwert in Scrum versus traditionellen Projekten

4.3.1 Verantwortlichkeiten des Product Owners in der Geschäftlichen Rechtfertigung

Die Verantwortung für die Priorisierung und Lieferung von Geschäftswerten der Projekte liegt hauptsächlich beim Product Owner. Für Programme und Portfolios liegt die Verantwortung beim Programm Product Owner beziehungsweise beim Portfolio Product Owner. Ihre Aufgabe ist es, als effektiver Repräsentant des Kunden und/oder Sponsors aufzutreten. Die Richtlinien zur Einschätzung und Bewertung des Geschäftswerts werden üblicherweise vom Scrum Guidance Body erstellt.

Abbildung 4-2 veranschaulicht die Verantwortlichkeiten der Geschäftlichen Rechtfertigung in einer hierarchischen Aufstellung.

Portfolio Product Owner	<ul style="list-style-type: none"> • Liefert Werte für Portfolios • Erstellt Geschäftliche Rechtfertigung für Portfolios • Bietet Wert-Richtlinien für Programme • Genehmigt Geschäftliche Rechtfertigung für Programme
Programm Product Owner	<ul style="list-style-type: none"> • Liefert Werte für Programme • Erstellt Geschäftliche Rechtfertigung für Programme • Bietet Wert Richtlinien für Projekte • Genehmigt Geschäftliche Rechtfertigung für Projekte
Product Owner	<ul style="list-style-type: none"> • Liefert Werte für Projekte • Erstellt Geschäftliche Rechtfertigung für Projekte • Bestätigt Realisierung der Werte für Business Stakeholdern

Abbildung 4-2: Hierarchie der Verantwortlichkeit der Geschäftlichen Rechtfertigung

4.3.2 Verantwortlichkeiten der anderen Scrum Rollen in der Geschäftlichen Rechtfertigung

Es ist wichtig zu vermerken, dass wenn auch der Product Owner die hauptsächliche Verantwortung für die Geschäftliche Rechtfertigung trägt, andere am Scrum Projekt arbeitende Personen folgende wesentliche Beiträge leisten:

1. Der **Sponsor** liefert die Finanzierung des Projekts und überwacht das Projekt fortlaufend um die Realisierung des Nutzens zu bestätigen.

2. **Kunden** und **Anwender** sind an der Festlegung der priorisierten Anforderungsliste und User Storys im Priorisierten Produkt Backlog beteiligt, sie bewerten zu liefernde Ergebnisse nach jedem Sprint oder Release und bestätigen, dass Nutzen erzielt wurde.
3. Der **Scrum Guidance Body** kann Richtlinien und Vorschläge in Bezug auf Techniken zur Geschäftlichen Rechtfertigung bieten, Wertrealisierungen bestätigen und so weiter. Auf solche Richtlinien und Vorschläge können sich das Scrum Kernteam und die Business Stakeholdern dann beziehen.
4. Der **Scrum Master** fördert die Erstellung der zu liefernden Ergebnisse des Projekts, managt Risiken, Änderungen und Hindernisse während des *Täglichen Standup durchführen*, *Sprint Rückblick halten* und anderer Scrum Prozesse. Der Scrum Master koordiniert mit dem Scrum Team, um die zu liefernden Ergebnisse zu erstellen, und mit dem Product Owner sowie anderen Business Stakeholdern um sicherzustellen, dass Nutzen durch das Projekt realisiert wird.
5. Das **Scrum Team** arbeitet daran, die zu liefernden Ergebnisse des Projekts zu erstellen und trägt seinen Teil dazu bei, einen Geschäftswert für alle Business Stakeholdern und das Projekt zu realisieren. Das Scrum Team ist auch an *Epics entwickeln*; *Priorisiertes Produkt Backlog erstellen*; *User Storys Erstellen*; *User Storys abschätzen*, *User Storys zusagen* und dazugehörigen Prozessen beteiligt, in denen die Geschäftlichen Anforderungen deutlicher definiert und priorisiert werden. Das Scrum Team hilft auch bei der Identifizierung von Risiken und reicht in Sprint Rückblick und anderen Meetings Change Requests zur Verbesserung ein.

4.4 Die Wichtigkeit der Geschäftlichen Rechtfertigung

Geschäftliche Rechtfertigung zeigt die Gründe für die Durchführung eines Projekts auf. Sie beantwortet die Frage 'Warum wird dieses Projekt benötigt?' Geschäftliche Rechtfertigung liefert den Antrieb für die gesamte Entscheidungsfindung in Bezug auf das Projekt. Daher ist es nicht nur wichtig, die Rentabilität und Ausführbarkeit eines Projekts zu bewerten, bevor wesentliche Ausgaben oder Investments zu Beginn des Projekts zugesagt werden, sondern auch, die Geschäftliche Rechtfertigung den Projektzyklus hindurch zu prüfen. Ein Projekt, das als unrentabel befunden wurde, sollte beendet werden; die Entscheidung sollte an die relevanten Business Stakeholdern und das Senior Management weitergereicht werden. Die Geschäftliche Rechtfertigung eines Projekts muss zu Beginn eines solchen beurteilt sein, und sowohl in vorher festgelegten Intervallen das Projekt hindurch, oder auch jederzeit, wenn größere Probleme oder Risiken auftauchen, die die Projekt-Ausführbarkeit bedrohen, wiederholt werden.

4.4.1 Faktoren um Geschäftliche Rechtfertigung zu bestimmen

Es gibt unzählige Faktoren die ein Product Owner bedenken muss, um die Geschäftliche Rechtfertigung eines Projekts zu bestimmen. Es folgen einige der wichtigsten Faktoren:

- **Projektbegründung**

Projektbegründung beinhaltet alle Faktoren, die das Projekt erforderlich machen, ob positiv oder negativ, gewählt oder nicht (z.B. ungenügende Kapazität zur Deckung des bestehenden und des prognostizierten Bedarfs, Abnahme der Kundenzufriedenheit, niedrige Erträge, gesetzliche Anforderungen usw.).

- **Geschäftsbedarf**

Geschäftsbedarf sind die vom Projekt zu erbringenden Geschäftsergebnisse, wie sie in der Projektvision dokumentiert sind.

- **Projektnutzen**

Der Projektnutzen umfasst alle messbaren Verbesserungen in einem Produkt, einer Dienstleistung oder einem Ergebnis, die durch den erfolgreichen Abschluss eines Projekts zur Verfügung gestellt werden könnten.

- **Gelegenheitskosten**

Gelegenheitskosten beziehen sich auf den Wert der nächstbesten Geschäfts-Option oder des nächstbesten Projektes, das zugunsten des ausgewählten Projekts verworfen wurde.

- **Große Risiken**

Risiken schließen alle Ungewissheiten oder nicht geplante Ereignisse ein, die die Durchführbarkeit und den potenziellen Erfolg des Projekts beeinflussen könnten.

- **Projekt-Zeitpläne**

Zeitpläne spiegeln die Länge oder Dauer eines Projekts wider und das Zeitintervall, über das hinweg der Nutzen des Projekts realisiert wird.

- **Projektkosten**

Die Projektkosten sind Investitionen und andere Entwicklungskosten, sowie die laufenden Kosten des Projekts, um die zu liefernden Ergebnisse zu pflegen.

4.4.2 Geschäftliche Rechtfertigung und der Lebenszyklus des Projekts

Die Geschäftliche Rechtfertigung wird zum ersten Mal vor dem Projektstart bewertet und wird kontinuierlich durch den Projekt Lebenszyklus hindurch überprüft. Folgende Maßnahmen zeigen auf, wie die Geschäftliche Rechtfertigung festgestellt wird:

1. Bewerten und vorstellen des Business Case

Die Geschäftliche Rechtfertigung für ein Projekt wird normalerweise vom Product Owner analysiert und bestätigt. Sie wird in Form eines Business Case vor dem Start des Projekts dokumentiert und vorgestellt und beinhaltet, die verschiedenen Faktoren aus Abschnitt 4.4.1 zu berücksichtigen. Sobald die Dokumentation erstellt wurde, sollte der Product Owner die Projektvisionserklärung entwerfen und dann die Zustimmung der Hauptentscheidungsträger der Organisation dazu einholen. In der Regel sind dies Führungskräfte und/oder irgendeine Art des Projekt- oder Programm Management Board.

2. Kontinuierliche Wertrechtfertigung

Nachdem die Entscheidungsträger die Projektvisionserklärung genehmigen, wird diese als Grundlage genommen und formt die Geschäftliche Rechtfertigung. Die Geschäftliche Rechtfertigung wird in der Projektausführung durchgehend validiert, üblicherweise in vordefinierten Intervallen oder Meilensteinen wie zum Beispiel während Portfolio, Programm und Produkt Priorisierten Backlog Review Meetings und wenn grössere Probleme und Risiken, die die Produkt Durchführbarkeit gefährden, identifiziert werden. Dies könnte in mehreren Scrum Prozessen geschehen, einschließlich *Tägliches Standup durchführen* und *Priorisiertes Produkt Backlog verfeinern*. Der Product Owner sollte die Geschäftliche Rechtfertigung in der Projektvisionserklärung das gesamte Projekt hindurch durchgehend mit relevanten Projekt-Informationen aktualisieren, um es den Hauptentscheidungsträgern zu ermöglichen, weiterhin sachkundige Entscheidungen zu treffen.

3. Realisierung des Nutzens bestätigen

Der Product Owner bestätigt das Erreichen von Nutzen für die Organisation während des gesamten Projektes und auch nach Vollendung von User Storys im Priorisierten Produkt Backlog. Nutzen aus Scrum Projekten wird während den *Sprint Vorführen und Validieren*, *Sprint Rückblick halten*, *Zu liefernde Ergebnisse ausliefern* und *Release Rückblick halten* Prozessen realisiert.

Abbildung 4-3 fasst die Schritte zur Bestimmung der Geschäftlichen Rechtfertigung zusammen.

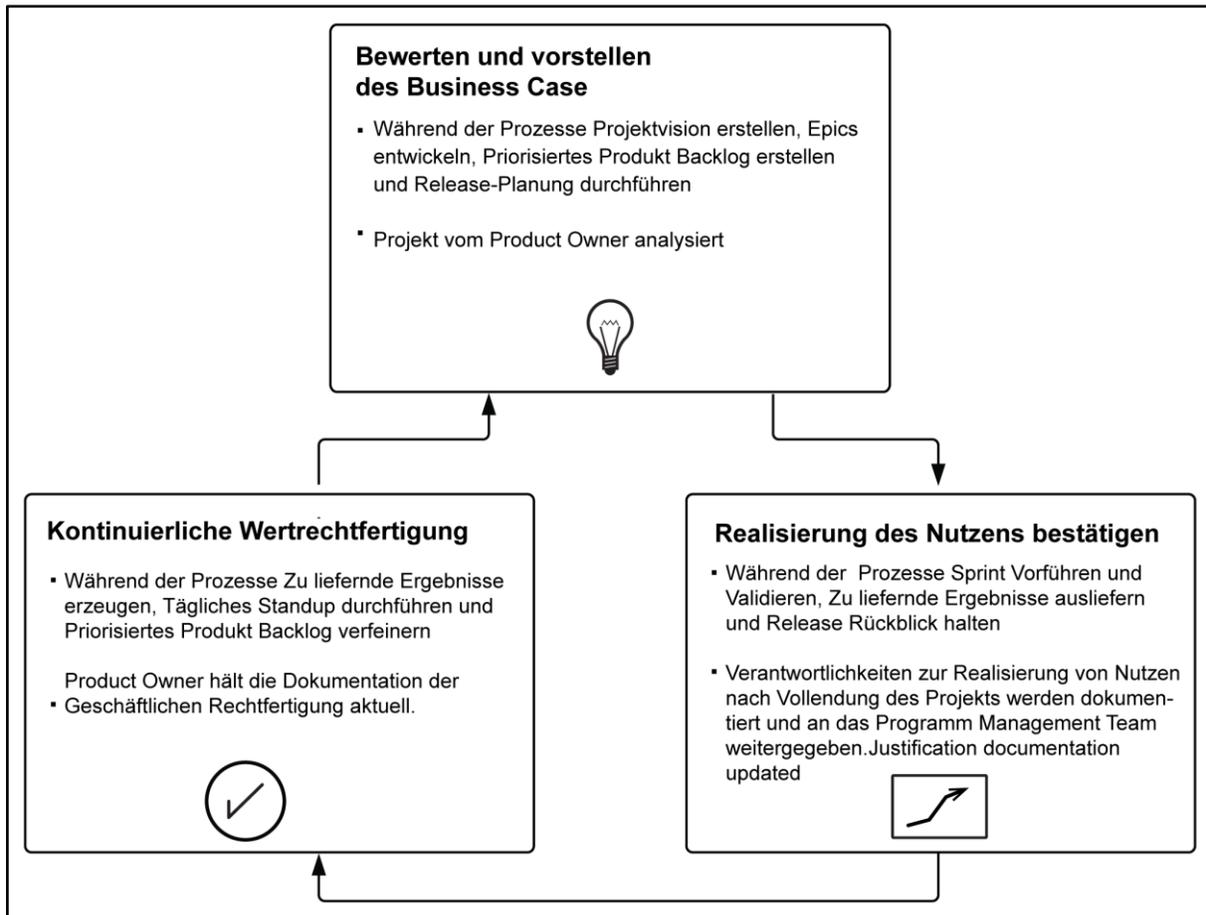


Abbildung 4-3: Geschäftliche Rechtfertigung und der Projekt Lebenszyklus

4.5 Techniken der Geschäftlichen Rechtfertigung

Der folgende Abschnitt behandelt einige der Hilfsmittel zur Bewertung und Auswertung der Geschäftlichen Rechtfertigung, sowie einige andere Aspekte, die mit der Projekt-Rechtfertigung und der Projekt-Auswahl verbunden sind. Es ist weder nötig noch empfehlenswert, jede verfügbare Technik für jedes Projekt anzuwenden. Manche dieser Techniken sind, abhängig vom genauen Projekt, nicht geeignet und andere Techniken können verwendet werden, um Projekte einzeln zu beurteilen oder den erwarteten Mehrwert mehrerer Projekte zu vergleichen.

Der Scrum Guidance Body (SGB) kann ein Gremium von Experten oder ein Satz Dokumente zu organisatorischen Standards und Verfahren sein und definiert die Richtlinien und Metriken, die zur Bestimmung des Geschäftswerts benutzt werden.

Jeder betreffende Product Owner ist dennoch dafür verantwortlich, die Tätigkeiten zur Verifizierung und Verfolgung des Geschäftswerts für seine oder ihre Projekte, Programme oder Portfolios durchzuführen.

4.5.1 Einschätzung des Projekt Mehrwerts

Der von Geschäftsprojekten zu liefernde Mehrwert kann durch Anwendung verschiedener Methoden wie Investitionsrendite (ROI), Kapitalwert (NPV) und Interner Zinsfuß (IRR) eingeschätzt werden.

1. Investitionsrendite (ROI)

Wenn für die Projektrechtfertigung verwendet, beurteilt die Investitionsrendite (ROI) erwarteten Nettoerträge, die aus einem Projekt gewonnen werden. Es wird berechnet, indem zunächst die zu erwartenden Kosten oder Investitionen in ein Projekt vom erwarteten Umsatz abgezogen werden. Dieser Nettogewinn wird dann durch die zu erwartenden Kosten dividiert, um die Ertragsrate zu bekommen. Andere Faktoren wie Inflation und Zinssätze für geliehenes Geld können in ROI Kalkulationen einbezogen werden.

ROI Formel:

$$\text{ROI} = (\text{Erwarteter Projektumsatz} - \text{Projektkosten}) / \text{Projektkosten}$$

Beispiel: Die ROI für ein Projekt, dessen Entwicklung \$125,000 kosten wird, mit erwarteten finanziellen Erträgen, die auf \$300,000 geschätzt werden, wird folgendermaßen kalkuliert:

$$\text{ROI} = (\$300.000 - \$125.000) / \$125,000 = 1,4$$

Demzufolge ist die ROI 1,4 mal die Investition (oder 140%).

Häufige Inkremente der Produkt oder Dienstleistungen sind eine entscheidende Grundlage von Scrum, die einen frühzeitigen Nachweis der ROI erlaubt. Dies hilft in der Bewertung der Rechtfertigung eines fortlaufenden Mehrwerts.

2. Kapitalwert (NPV)

Kapitalwert (NPV) ist eine Methode zur Bestimmung des aktuellen Kapitalwerts eines zukünftigen finanziellen Vorteils, basierend auf einer angenommenen Inflationsrate oder einem angenommenen Zinssatz. In anderen Worten ist NPV der gesamte erwartete Ertrag oder Umsatz eines Projekts, abzüglich der gesamten erwarteten Projektkosten, der Zeitwert des Geldes miteinberechnet.

Beispiel: Welche der folgenden zwei Projekte sollte ausgewählt werden wenn NPV als Auswahl Kriterium benutzt wird?

1. Projekt A hat einen NPV von \$1.500 und wird in 5 Jahren fertiggestellt sein.
2. Projekt B hat einen NPV von \$1.000 und wird in 1 Jahr fertiggestellt sein.

Lösung: Projekt A, denn der NPV ist höher; der Fakt das Projekt B eine kürzere Laufzeit als Projekt A hat wird hierbei nicht berücksichtigt, denn Zeit wurde in der NPV Kalkulation schon eingerechnet (d.h., aufgrund der Tatsache, dass der derzeitige, nicht der zukünftige Wert in der Kalkulation berücksichtigt wird).

3. Interner Zinsfuß (IRR)

Interner Zinsfuß (IRR) ist ein Zinssatz für eine Investition, bei dem der gegenwärtige Wert der Bargeldzuflüsse dem gegenwärtigen Wert der Bargeldabflüsse gleichgesetzt wird, um den Zinsfuß des Projekts zu ermitteln. Beim Vergleich von Projekten ist eines mit einem höheren IRR in der Regel besser.

Obwohl IRR nicht so oft zur Rechtfertigung von Projekten benutzt wird wie andere Techniken, wie zum Beispiel NPV, ist er dennoch ein wichtiges Konzept, das man kennen sollte.

Beispiel: Basierend auf IRR, welches Projekt ist erstrebenswerter?

1. Projekt A hat einen IRR von 15% und wird in 5 Jahren abgeschlossen sein.
2. Projekt B hat einen IRR von 10% und wird in 1 Jahr abgeschlossen sein.

Lösung: Projekt A, denn sein IRR ist größer; der Fakt das Projekt B eine kürzere Laufzeit als Projekt A hat wird hierbei nicht berücksichtigt, denn Zeit wurde in den IRR Kalkulationen schon eingerechnet (d.h., wie bei NPV, der derzeitige, nicht der zukünftige Wert wird benutzt um den IRR zu berechnen).

4.5.2 Planung für Mehrwert

Nachdem der Projektwert gerechtfertigt und bestätigt wurde, sollte der Product Owner die vom Scrum Guidance Body (oder einem ähnlichen organisatorischen Projektboard oder Funktion) vorgeschriebenen organisatorischen Richtlinien, Verfahren, Vorlagen und Allgemeine Standards bei der Planung eines Projekts berücksichtigen; gleichzeitig die wertgetriebene Lieferung maximieren. Die Beweislast zur Bestimmung wie der Mehrwert erzeugt wird, fällt auf die Business Stakeholdern (Sponsor, Kunden und/oder Anwender), während das Scrum Team sich darauf konzentriert, was zu entwickeln ist. Einige verbreitete Hilfsmittel, die vom Scrum Guidance Body empfohlen werden, könnten folgende einschließen:

1. Value Stream Mapping

Value Stream Mapping verwendet Flussdiagramme, um den Fluss der für einen Prozess erforderlichen Schritte zu veranschaulichen. Diese Technik kann zur Optimierung eines Prozesses eingesetzt werden, indem sie hilft, nicht wertschöpfende Elemente zu identifizieren und zu eliminieren und die Effizienz zu steigern. Value Stream Mapping kann zur Optimierung von Scrum-Prozessen verwendet werden, um beispielsweise die Sprint-Geschwindigkeit zu verbessern.

Abbildung 4-4 zeigt, wie die Identifizierung von Prozess- und Wartezeiten zur Optimierung des Systems beitragen kann, indem die Wartezeiten verringert und die Prozesseffizienz verbessert werden.

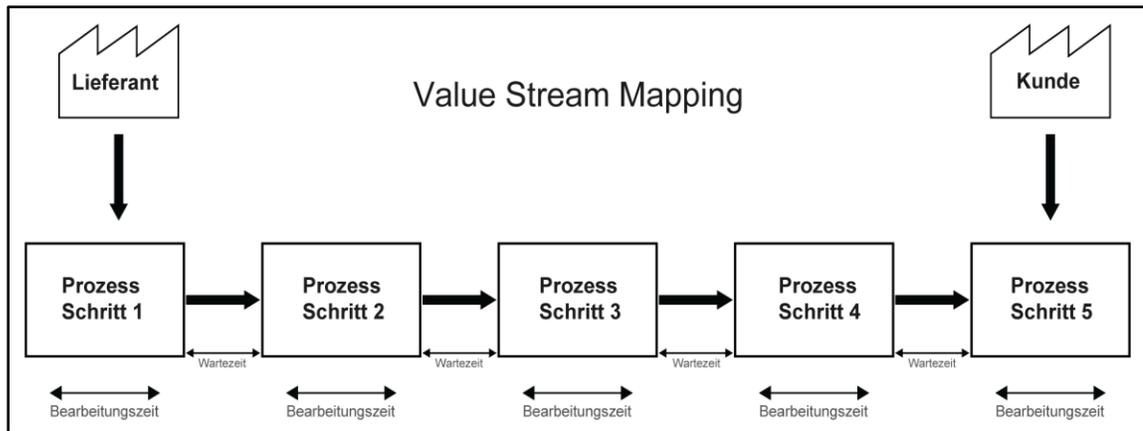


Abbildung 4-4: Value Stream Mapping

2. Kundenwertorientierte Priorisierung

Kundenwertorientierte Priorisierung legt größten Wert auf den Kunden und ist bestrebt, User Stories mit dem höchsten Wert zuerst zu implementieren. Die User Stories mit dem höchsten Wert werden identifiziert und an den Anfang des Priorisierten Produkt Backlogs bewegt.

Ein Team kann eine Vielfalt an Schemata zur Priorisierung benutzen, um hochwertige Features zu ermitteln.

a. Einfache Schemata

Einfache Schemata involvieren die Kennzeichnung von Gegenständen als „1“, „2“, „3“ oder „Hoch“, „Mittel“ und „Niedrig“ und so weiter. Obwohl dies eine einfache und unkomplizierte Vorgehensweise ist, kann sie problematisch werden, da es oft eine Tendenz gibt, alles als Priorität „1“ oder „Hoch“ zu kennzeichnen. Selbst „Hoch“, „Mittel“, und „Niedrig“ Priorisierungsschemata können ähnliche Schwierigkeiten mit sich bringen.

b. MoSCoW-Priorisierung

Der Begriff MoSCoW-Priorisierung leitet seinen Namen von den Anfangsbuchstaben der Formulierungen „Must have“, „Should have“, „Could have“ und „Would like to have, but not at this time“ ab. Diese Methode der Priorisierung ist generell effektiver als die einfacher Schemata. Diese legen Prioritäten in absteigender Reihenfolge fest, wobei sich „Must have“ Features so definieren, dass das Produkt ohne sie wertlos ist, und „Would like to have“ Features für jene Eigenschaften stehen, die zwar nett wären, jedoch nicht zwingend inbegriffen sein müssen.

c. Monopoly Geld

Technik, bei der dem Kunden „Spiel-“ bzw. „Monopoly-Geld“ in der Höhe des tatsächlichen Projektbudgets mit der Bitte ausgehändigt wird, dieses auf die in Betracht gezogenen User Storys zu verteilen. Auf diese Weise priorisiert der Kunde mittels seiner Zahlungsbereitschaft für jeder einzelne User Story.

d. 100-Punkte-Methode

Die 100-Punkte-Methode wurde von Dean Leffingwell und Don Widrig (2003) entwickelt. Sie sieht vor, dem Kunden 100 Punkte zur Verfügung zu stellen, mit denen dieser die für ihn wichtigsten Features bewerten kann.

e. Kano-Analyse

Kano-Analyse wurde von Noriaki Kano (1984) entwickelt und betrifft die Klassifizierung von Features oder Anforderungen in vier Kategorien, die auf der Grundlage von Kundenpräferenzen definiert werden:

1. *Begeisterungs-Features*: Features, die für den Kunden neu oder von hohem Wert sind
2. *Zufriedenheits-Features*: Features, die dem Kunden Wert bieten
3. *Ablehnungs-Features*: Features, die, wenn nicht vorhanden, den Kunden sehr wahrscheinlich dazu bringen werden, das Produkt abzulehnen, aber den Zufriedenheitslevel nicht beeinflussen, wenn sie vorhanden sind
4. *Unerhebliche-Features*: Features, die den Kunden in keiner Weise beeinträchtigen werden und aus dem Projekt herausgenommen werden sollten

Abbildung 4-5 zeigt eine bildliche Veranschaulichung der Kano-Analyse.

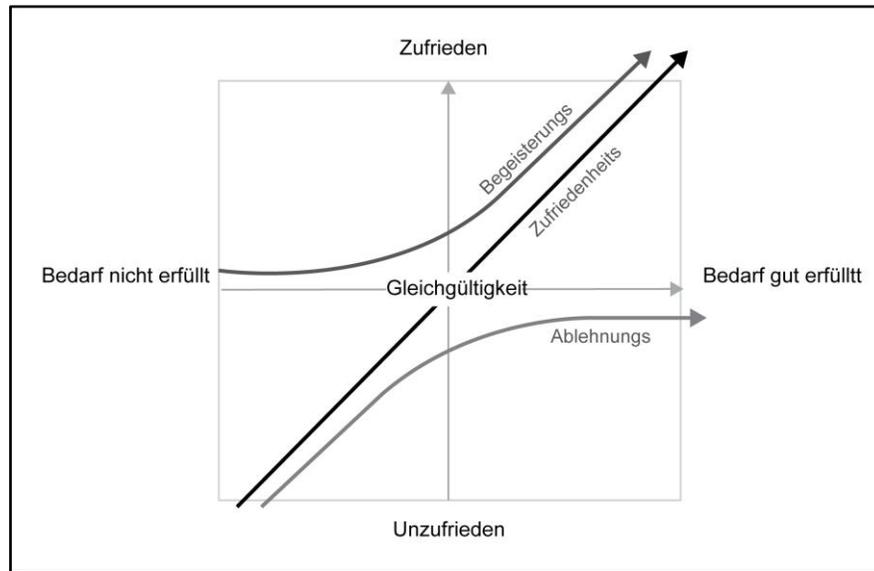


Abbildung 4-5: Kano-Analyse

Interessanterweise bewegen sich Features normalerweise mit der Zeit die Einstufungsleiter hinunter; Kunden werden Features erwarten (z.B. Kameras in Telefonen) und diese Features bewegen sich von Begeisterungs- zu Zufriedenheits- und schlussendlich zu Ablehnungsbereichen.

4.5.3 Relative Priorisierungsrangliste

Eine einfache Auflistung der User Storys nach Priorität. Es ist eine wirksame Methode zur Bestimmung der gewünschten User Storys für die jeweilige Iteration oder das jeweilige Release des Produkts oder der Dienstleistung. Der Zweck ist eine einfache, einzelne Liste zu erstellen mit dem Ziel, Features zu priorisieren, anstatt von mehreren Priorisierungs-Schemata abgelenkt zu werden.

Diese einfache Liste liefert außerdem die Grundlage, Änderungen einzubauen und wenn nötig, Risiken zu identifizieren. Alle Änderungen oder identifizierte Risiken können in die Liste eingefügt werden, basierend auf ihrer Priorität in Relation zu den anderen User Storys der Liste. Üblicherweise werden neue Änderungen auf Kosten von Features eingeführt, die eine niedrigere Priorität haben.

Den Minimalen Vermarktbareren Funktionsumfang (MMF, Minimum Marketable Featureset) zu definieren ist äußerst wichtig in diesem Prozess, sodass das erste Release oder die Iteration so früh wie möglich geschieht, was zu einer erhöhten ROI führt. Normalerweise müssten diese User Storys die höchstpriorären in der Priorisierungsrangliste sein.

4.5.4 Story Mapping

Story Maps stellen die Abfolge der Iterationen der Produktentwicklung dar und zeigen auf, welche User Storys in der ersten, zweiten, dritten und folgenden Versionen enthalten sein werden. Bei der Anwendung von Scrum-Praktiken ist es selbstverständlich, dass es sich hierbei nur um einen aktuellen Ausblick handelt, der regelmäßig überprüft wird und sich häufig ändert.

Das Beispiel der Story Map in Abbildung 4-6 veranschaulicht, wie das Scrum-Team verschiedene Releases plant und dem Release, das in naher Zukunft erscheint, eine höhere Priorität zuweist. Es wird erwartet, dass das Team ein besseres Verständnis für die User Storys im nächsten Release hat; je weiter ein Release in der Zukunft liegt, desto wahrscheinlicher ist es, dass sich noch Änderungen ergeben.

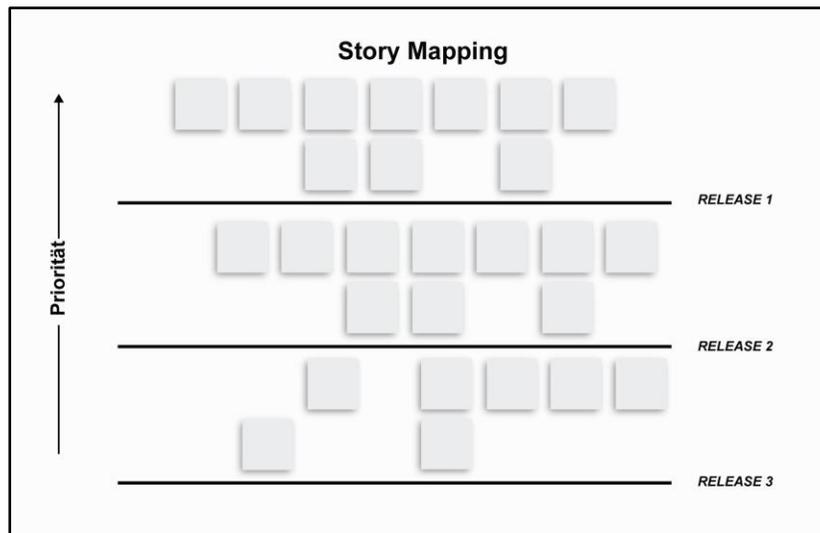


Abbildung 4-6: Story Mapping

4.6 Kontinuierliche Wertrechtfertigung

Der Geschäftswert sollte regelmäßig bemessen werden, um festzustellen, ob die Rechtfertigung oder Rentabilität, das Projekt auszuführen, weiterhin besteht. Häufige Bewertung der Investitionen in das Projekt in Relation zu dem erstellten Geschäftswert qualifiziert die fortlaufende Rentabilität eines Projekts. Die erwarteten Anforderungen an das Projekt mögen sich häufig ändern, was sowohl auf die Projekt-Investition als auch auf die Wertschöpfung Auswirkungen haben kann. Ein entscheidender Scrum Aspekt ist seine Fähigkeit, sich schnell an durch ein rasch verändertes Geschäftsmodell erzeugtes Chaos anzupassen. In Projekten mit mehrdeutigen Anwender Anforderungen und erheblichem Potential für häufige Änderungen bietet Scrum beträchtliche Vorteile gegenüber anderen Entwicklungsmodellen.

Die Rate der Wertlieferung zu überwachen ist eine wichtige Anforderung an Scrum Projekte. Periodisch die Erstellung von Mehrwerten zu verfolgen und zu berichten hilft in der Bewertung des Projektstatus und liefert wichtige Informationen an den Kunden und andere Business Stakeholdern.

4.6.1 Ertragswertanalyse

Auch wenn Hilfsmittel wie Balkendiagramme und Gantt Charts häufig benutzt werden, haben sie Einschränkungen beim Verfolgen und Berichten des Fortschritts, wenn es um die Projektleistung geht. Die Ertragswertanalyse (EVA) wird für diesen Zweck benutzt. Ertragswertanalyse analysiert die aktuelle Projektleistung gegen die geplante Leistung zu einem bestimmten Zeitpunkt. Damit Techniken zur Nachverfolgung effektiv bleiben, muss der anfängliche Grundplan des Projekts akkurat sein. EVA benutzt oft Grafiken und andere visuelle Hilfsmittel (z.B. S-Kurve), um die Projektstatus Informationen dazustellen.

Die Ertragswertanalyse misst aktuelle Abweichungen im Projekt-Zeitplan und die Kostenleistung und prognostiziert die endgültigen Kosten auf der Grundlage der ermittelten aktuellen Leistung. EVA wird üblicherweise am Ende jeden Sprints durchgeführt, nachdem die User Storys im Sprint Backlog fertiggestellt wurden.

Tabelle 4-1 fasst die benutzten Formeln der Ertragswertanalyse zusammen.

Begriff Definition	Akronym	Formel
Geplanter Wert	PV	
Fertigstellungswert	EV	
Ist-Kosten	AC	
Geplante Kosten	BAC	
Terminplanabweichung	SV	$EV - PV$
Kostenabweichung	CV	$EV - AC$
Terminentwicklungsindex	SPI	EV / PV
Kostenentwcklungsindex	CPI	EV / AC
Prozent Abgeschlossen	% Complete	$(EV / BAC) \times 100$
Erwartete Kosten zur Fertigstellung <ul style="list-style-type: none"> • Kalkulierte Annahmen nicht gültig • Derzeitige Abweichungen sind atypisch • Derzeitige Abweichungen sind typisch 	EAC	<ol style="list-style-type: none"> 4. $AC + ETC$ 5. $AC + BAC - EV$ 6. BAC / CPI
Erwartete Restkosten zur Fertigstellung	ETC	$EAC - AC$
Gesamtkostenabweichung bei Abschluss	VAC	$BAC - EAC$

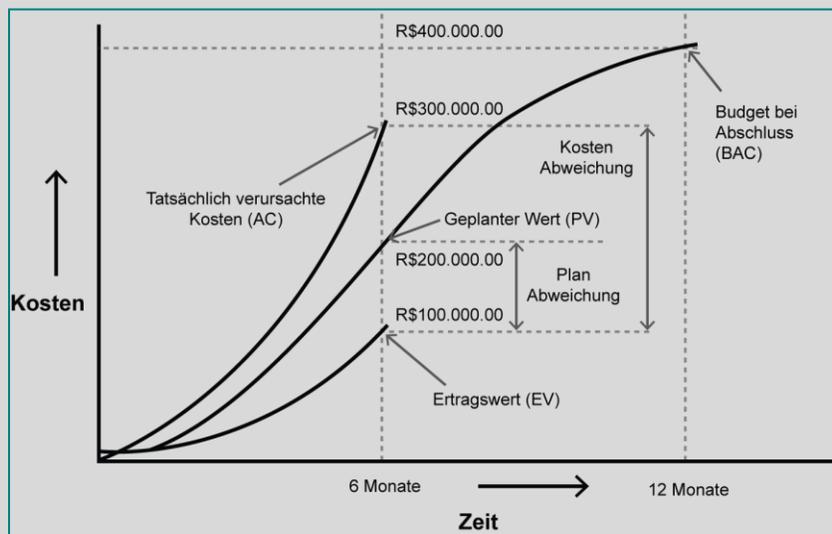
Tabelle 4-1: Ertragswert Formeln

Beispiel: Eine Website mit 4.000 Webseiten muss entwickelt werden—wir gehen davon aus, dass die Erstellung jede Webseite die gleiche Zeit benötigt und dass jede Webseite eine einzigartige User Story mit gleichwertiger Priorität im Priorisierten Produkt Backlog darstellt. Die geschätzten Kosten zur Fertigstellung des Projekts belaufen sich auf \$400.000 und die zeitliche Begrenzung für das Projekt liegt bei 12 Monaten. Nach 6 Monaten sind \$300.000 ausgegeben worden und 1.000 Webseiten wurden erstellt.

Was hat man uns geliefert?

- Geplante Kosten (BAC) = \$400.000 (Grundwert der Projektkosten)
- Geplanter Wert (PV) = \$200.000 (da wir mit 2.000 Webseiten geplant hatten)
- Fertigstellungswert (EV) = \$100.000 (Wert der 1.000 fertiggestellten Webseiten)
- Ist-Kosten (AC) = \$300.000 (was bis jetzt ausgegeben wurde)

S-Kurve für die Daten:



Formeln:

- Terminplanabweichung (SV) = $EV - PV = \$100.000 - \$200.000 = - \$100.000$
- Kosten Abweichung (CV) = $EV - AC = \$100.000 - \$300.000 = - \$200.000$
 - Die negative Abweichung in unserem Projekt deutet an, dass wir dem Plan hinterher und über dem Budget sind.
- Terminentwicklungsindex (SPI) = $EV / PV = \$100.000 / \$200.000 = 0,5$
 - SPI < 1 deutet an das die bis jetzt erledigte Arbeit nur 50% von dem ist was wir für die 6 Monate geplant hatten fertigzustellen.
- Kostenentwicklungsindex (CPI) = $EV / AC = \$100.000 / \$300.000 = 0,33$
 - CPI < 1 deutet an, dass wir nur 33% der Arbeit erledigen in Bezug auf das ausgegebene Geld.
- Prozent Abgeschlossen = $EV / BAC \times 100 = \$100.000 / \$400.000 \times 100 = 25\%$
 - Nur 25% der Projekt Arbeit ist zu diesem Zeitpunkt abgeschlossen.

4.6.2 Kumulatives Flussdiagramm (CFD)

Ein Kumulatives Flussdiagramm (CFD) ist ein nützliches Hilfsmittel für die Berichterstattung über und die Verfolgung der Projekt Performance. Es bietet eine einfache, visuelle Abbildung des Projektfortschritts zu einem bestimmten Zeitpunkt. Es wird normalerweise dazu verwendet, um einen High-Level Status des Gesamtprojekts bereitzustellen. Es ist nicht dazu da, tägliche Aktualisierungen für einzelne Sprints zu liefern.

Abbildung 4-7 ist ein Beispiel eines CFD für ein Großprojekt. Es zeigt, wie viele User Storys noch erstellt werden müssen, gegenwärtig erstellt werden und bereits erstellt wurden. Wenn Kundenanforderungen sich ändern, ergibt dies eine Änderung in den kumulativen User Storys, die geliefert werden müssen. Änderungspunkte 1 und 2 ergeben sich, wenn der Product Owner existierende User Storys aus dem Risiko angepassten Priorisierten Produkt Backlog entfernt und Änderungspunkte 3 und 4 ergeben sich, wenn der Product Owner neue User Storys in das Risiko angepasste Priorisierte Produkt Backlog einfügt.

Diese Art von Diagramm kann ein großartiges Hilfsmittel zur Identifizierung von Hindernissen und Engpässen in den Prozessen sein. Wenn zum Beispiel das Diagramm zeigt, dass ein Bereich schmaler wird, während das vorige mit der Zeit breiter wird, könnte dies einen Engpass darstellen und Änderungen mögen nötig sein, um die Effizienz zu vergrößern.

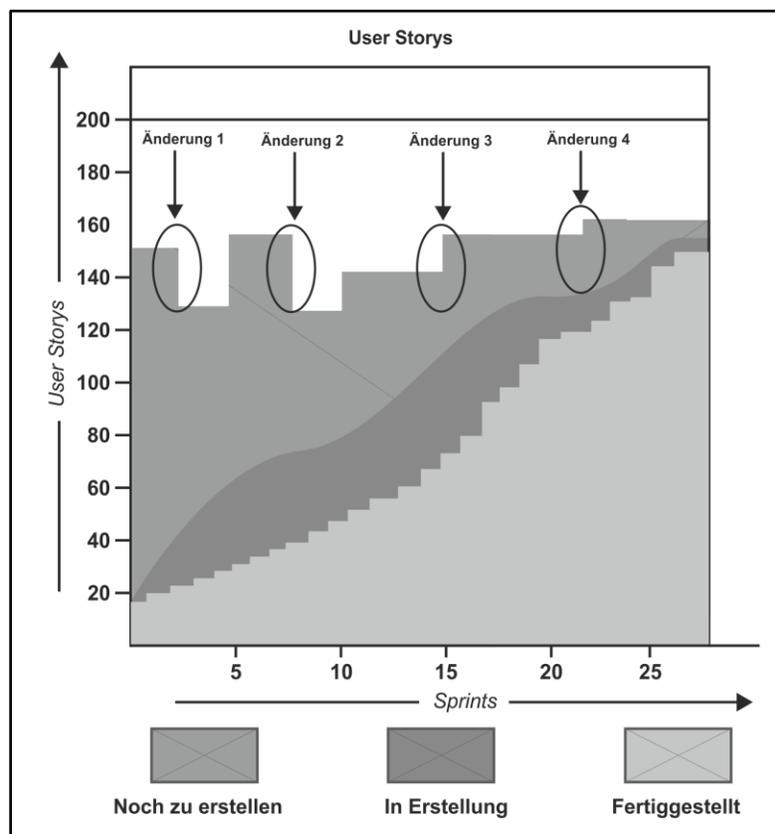


Abbildung 4-7: Beispiel Kumulatives Flussdiagramm (CFD)

4.7 Bestätigen der Ertragsrealisierung

Durch das gesamte Projekt hindurch ist es wichtig nachzuprüfen, ob Nutzen erzielt wird. Ob die Produkte eines Scrum Projektes greifbar oder nicht greifbar sind, geeignete Verifikationstechniken werden benötigt, um zu bestätigen, dass das Team die zu liefernden Ergebnisse erzeugt, welche den zu Beginn des Projekts festgelegten Nutzen und Mehrwert erreicht.

4.7.1 Prototypen, Simulationen und Vorführungen

Den Kunden Prototypen vorzustellen und deren Funktionsweisen zu simulieren sind allgemein übliche Techniken, um den Mehrwert zu bestätigen.

Oftmals, nachdem die Features angewendet oder vorgeführt wurden, können die Kunden deutlicher sehen, ob die Features angemessen sind und zu ihren Bedürfnissen passen. Sie mögen feststellen, dass zusätzliche Features erforderlich sind oder sich entschließen vorher definierte Feature-Anforderungen zu modifizieren. In der Produktentwicklung wird dieses Kundenerlebnis als IKIWISI bezeichnet (I'll Know It When I See It / Ich weiß es, wenn ich es sehe).

Durch Vorführungen oder Zugang zu frühen Iterationen können Kunden außerdem bewerten, in welchem Maße das Team ihre Anforderungen verstanden und ihre Erwartungen erfüllt hat.

4.8 Zusammenfassung der Verantwortlichkeiten

Rollen	Verantwortlichkeiten
Scrum Guidance Body	<ul style="list-style-type: none"> • Erstellt allgemeine Richtlinien und Metriken zur Bewertung von Mehrwert • Handelt in einer Beratungsfunktion und liefert nach Bedarf Richtlinien für Projekte, Programme und Portfolios
Portfolio Product Owner	<ul style="list-style-type: none"> • Stellt Wertlieferung für Portfolios sicher • Erstellt die Geschäftliche Rechtfertigung für Portfolios • Liefert Mehrwert Anleitung für Programme in einem Portfolio • Genehmigt die Geschäftliche Rechtfertigung von Programmen in einem Portfolio
Portfolio Scrum Master	<ul style="list-style-type: none"> • Stellt sicher, dass das gewünschte Ergebnis eines Portfolios erreicht wird • Vollzieht die Kontinuierliche Wertrechtfertigung für Portfolios
Programm Product Owner	<ul style="list-style-type: none"> • Stellt Wertlieferung für Programme sicher • Erstellt die Geschäftliche Rechtfertigung für Programme • Liefert Mehrwert Anleitung für Projekte in einem Programm • Genehmigt die Geschäftliche Rechtfertigung von Projekten in einem Programm
Programm Scrum Master	<ul style="list-style-type: none"> • Stellt sicher, dass die gewünschten Ergebnisse eines Programms bekannt sind und verstanden wurden • Vollzieht die Kontinuierliche Wertrechtfertigung für Programme
Business Stakeholdernn	<ul style="list-style-type: none"> • Hilft User Storys und Anforderungen im Priorisierten Produkt Backlog zu priorisieren • Kommuniziert mit dem Scrum Team und bestätigt am Ende jedes Sprints, Release und des Projekts die Realisierung von Mehrwert
Product Owner	<ul style="list-style-type: none"> • Stellt Wertlieferung für Projekte sicher • Erhält die Geschäftliche Rechtfertigung für Projekte • Bestätigt den Projektnutzen und vermittelt ihn an Business Stakeholdern
Scrum Master	<ul style="list-style-type: none"> • Stellt sicher, dass die gewünschten Ergebnisse eines Projekts bekannt sind und vom Scrum Team verstanden wurden • Vollzieht die Kontinuierliche Wertrechtfertigung für Projekte
Scrum Team	<ul style="list-style-type: none"> • Stellt sicher, dass zu liefernde Ergebnisse des Projekts in Übereinstimmung mit den abgemachten Abnahmekriterien fertiggestellt werden • Vollzieht die Kontinuierliche Wertrechtfertigung für Projekte

Tabelle 4-2: Zusammenfassung der Verantwortlichkeiten in Bezug zur Geschäftlichen Rechtfertigung

4.9 Scrum versus traditionelles Projekt Management

Traditionelle Projekte legen Wert auf umfangreiche Vorausplanung und die Einhaltung des vom Projektmanager entwickelten Projektplans. Im Allgemeinen werden Änderungen durch ein formelles Änderungsmanagement System gehandhabt und der Mehrwert wird am Ende des Projekts erzielt, wenn das Endprodukt geliefert wird.

In Scrum Projekten wird keine umfangreiche, langfristige Planung vor der Projektdurchführung durchgeführt. Die Planung wird schrittweise vor jedem Sprint ausgeführt. Dies ermöglicht eine schnelle und effektive Reaktion auf Änderungen, was zu niedrigeren Kosten führt und schlussendlich die Rentabilität und Investitionsrendite (ROI) erhöht. Zudem ist die wertgetriebene Lieferung (Abschnitt 4.3) einer der wichtigsten Vorteile des Scrum Rahmens und bietet wesentlich bessere Priorisierung und schnellere Verwirklichung von Geschäftswerten. Dank der iterativen Art der Scrum Entwicklung ist immer wenigstens ein Release des Produkts mit Minimalem Vermarktbarem Funktionsumfang (MMF) verfügbar. Auch wenn ein Projekt abgebrochen wird, wurden normalerweise schon vor dem Abbruch einiger Nutzen oder Werte erzeugt.

5. QUALITÄT

5.1 Einleitung

Das Ziel dieses Kapitels ist es Qualität in Bezug auf Projekte zu definieren und den Scrum Ansatz zum Erreichen des erforderlichen Maßes an Qualität vorzustellen,.

Qualität, wie im *Scrum Body of Knowledge (SBOK®)* definiert, ist auf Folgendes anwendbar:

- Portfolios, Programme und/oder Projekte in jeder Branche
- Produkte, Dienstleistungen oder jegliche an Business Stakeholdern zu liefernde Ergebnisse
- Projekte jeglicher Größe oder Komplexität

Der Begriff "Produkt" in dieser Wissenssammlung kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann auf jedes Projekt in jeder Branche effektiv angewendet werden — von kleinen Projekten oder Teams mit lediglich sechs Team-Mitgliedern bis hin zu großen, komplexen Projekten mit mehreren Hundert Team-Mitgliedern.

Dieses Kapitel ist in folgende Abschnitte aufgeteilt:

5.2 Leitfaden für die Kernrollen—Dieser Teil klärt die Frage, welche Abschnitte für jede Scrum Rolle relevant sind: Product Owner, Scrum Master und Scrum Team.

5.3 Qualität definiert—Dieser Abschnitt enthält die Scrum Definition von Qualität, mit einer klaren Abgrenzung gegenüber dem Umfang und beschreibt das Verhältnis von Qualität und dem geschäftlichen Mehrwert.

5.4 Abnahmekriterien und das Priorisierte Produkt Backlog—Dieser Abschnitt betont die Bedeutung der Abnahmekriterien, des Priorisierten Produkt Backlogs, und deren Verhältnis. Außerdem wird die Scrum Definition von Done erklärt.

5.5 Qualitätsmanagement in Scrum—Dieser Abschnitt enthält Details über Qualitätsplanung, Qualitätskontrolle und Qualitätssicherung im Zusammenhang mit Scrum.

5.6 Zusammenfassung der Verantwortlichkeiten—Dieser Abschnitt beschreibt die Verantwortungen jeder Person oder Rolle eines Projekts hinsichtlich Qualität.

5.7 Scrum versus traditionelles Projekt Management—Dieser Abschnitt hebt die Vorzüge des Qualitätsmanagements nach der Scrum Methode gegenüber den traditionellen Projekt Management Modellen hervor.

5.2 Leitfaden für die Kernrollen

1. Product Owner—Für jeden der die Rolle des Product Owners in Scrum Projekten übernimmt ist es wichtig diese Kapitel komplett zu lesen.
2. Scrum Master—Der Scrum Master sollte mit diesem kompletten Kapitel vertraut sein, besonders mit den Abschnitten 5.3, 5.4, 5.5.3, und 5.6.
3. Scrum Team—Das Scrum Team sollte sich hauptsächlich auf die Abschnitte 5.3, 5.4, und 5.6 konzentrieren.

5.3 Qualitätsdefinition

Es gibt unzählige Möglichkeiten Qualität zu definieren

Scrum definiert Qualität als die Fähigkeit des fertigen Produkts oder der zu liefernden Ergebnisse, die Abnahmekriterien zu erfüllen und den vom Kunden erwarteten geschäftlichen Wert zu erzielen.

Um sicher zu stellen, dass ein Projekt den Qualitätsvorgaben entspricht, wählt Scrum einen Ansatz der kontinuierlichen Verbesserung, bei dem das Team aus eigener Erfahrung und dem Engagement der Business Stakeholdern lernt, um das Priorisierte Produkt Backlog kontinuierlich bezüglich aller Änderungen auf dem aktuellen Stand zu halten. Das Priorisierte Produkt Backlog ist einfach niemals vollständig bis zur Schließung oder Beendigung des Projekts. Jegliche Änderung an den Vorgaben spiegelt Änderungen im internen und externen Geschäftsumfeld wieder und erlaubt dem Team kontinuierlich Anpassungen vorzunehmen, um alle Anforderungen zu erfüllen. Die Tatsache dass Scrum, durch wiederholtes Testen, ein stufenweises Arbeiten mit Sprints benötigt, anstatt bis zum Ende zu warten um Ergebnisse zu liefern, führt dazu, dass Fehler sofort behoben werden, anstatt aufgeschoben zu werden. Darüber hinaus werden wichtige, qualitätsrelevante Aufgaben (z.B. Entwicklung, Prüfung und Dokumentation) als Teil desselben Sprints vom selben Team erledigt - dies stellt die vorgegebene Qualität für jedes als Teil eines Sprints gelieferte Ergebnis sicher, das die Done Kriterien erfüllt. Die kontinuierliche Verbesserung optimiert somit durch wiederholtes Testen die Wahrscheinlichkeit das erwartete Maß an Qualität in einem Scrum Projekt zu erreichen. Regelmäßige Gespräche zwischen dem Scrum Kernteam und den Business Stakeholdern (Kunden und Anwender eingeschlossen) sowie das Bereitstellen von aktuellen Inkrementen des Produkts am Ende jedes Sprints stellen sicher, dass die Lücke zwischen Kundenerwartungen bezüglich des Projekts und den tatsächlich gelieferten Ergebnissen kontinuierlich verkleinert wird.

5.3.1 Qualität und Umfang

Umfang und Qualitätsvoraussetzungen für ein Projekt werden unter Berücksichtigung verschiedener Faktoren bestimmt, z.B.:

- Der Geschäftsbedarf, welchen das Projekt erfüllen wird

- Die Fähigkeit und Bereitschaft der Organisation, den identifizierten Geschäftsbedarf zu erfüllen
- Den jetzigen und zukünftigen Bedarf der Zielgruppe

Der Umfang des Projekts ist die Gesamtsumme aller Produktschritte und der für die Entwicklung des Endprodukts benötigten Arbeit. Qualität ist die Fähigkeit der zu liefernden Ergebnisse, die Qualitätsanforderung an das Produkt und die Kundenerwartungen zu erfüllen. In Scrum werden der Umfang und die Qualität des Projektes im Priorisierten Produkt Backlog erfasst und der Umfang eines jeden Sprints wird ermittelt durch Verfeinerung der umfangreichen Priorisierten Produkt Backlog Elemente (PBIs) in eine Menge kleiner, aber detaillierter User-Stories, die innerhalb eines Sprint geplant, entwickelt und verifiziert werden können.

Das Priorisierte Produkt Backlog wird kontinuierlich vom Product Owner gepflegt. Der Product Owner stellt sicher, dass alle User Storys, von denen erwartet wird, dass das Scrum Team sie in einem Sprint entwickelt, vor dem Start des Sprint verfeinert werden. Generell sind die wichtigsten Anforderungen zur Lösung der Kundenprobleme oder die Erfüllung ihrer Bedürfnisse hoch priorisiert und die restlichen werden niedriger eingestuft. Weniger wichtige User Storys werden in nachfolgenden Sprints entwickelt oder können je nach Kundenwunsch ganz ausgelassen werden. Während der Sprint Durchführung können der Product Owner, der Kunde und das Scrum Team die Merkmale des Produktes besprechen, um den sich ändernden Bedürfnissen des Kunden nachzukommen. Während des gesamten Projekts besprechen und ändern der Product Owner, Kunden und das Scrum-Team die Liste der Merkmale des Produkts, um den sich ändernden Bedürfnissen der Kunden gerecht zu werden.

5.3.2 Qualität und Geschäftswert

Qualität und Geschäftswert hängen eng zusammen. Demnach ist es entscheidend, die Qualität und den Umfang eines Projektes zu verstehen, um die Ergebnisse und die Vorteile korrekt auszuarbeiten, welche das Projekt und das Produkt erreichen müssen, um den Geschäftswert zu liefern. Um den Geschäftswert eines Produktes zu ermitteln ist es wichtig, den Geschäftsbedarf, welcher die Anforderungen an das Produkt erzeugt, zu kennen. Somit bestimmt der Geschäftsbedarf das benötigte Produkt und das Produkt wiederum liefert den erwarteten Geschäftswert.

Qualität ist eine komplexe Variable. Eine Vergrößerung des Projekt-Umfangs ohne Erhöhung der Dauer oder Ressourcen neigt dazu, die Qualität zu verringern. Ebenso führt eine Verringerung der Dauer oder der Ressourcen ohne Reduktion des Umfangs generell zu einer verminderten Qualität. Scrum glaubt an die Erhaltung eines „Nachhaltigen Arbeitstempos“, was hilft, die Qualität nach und nach zu verbessern.

Der Scrum Leitfaden kann Mindestqualitätsanforderungen und Standardanforderungen für alle Projekte in der Organisation definieren. Alle Scrum Teams der Firma müssen diese Standards einhalten.

5.4 Abnahmekriterien und das Priorisierte Produkt Backlog

Das Priorisierte Produkt Backlog ist das einzige Anforderungsdokument, welches den Projektumfang definiert, in dem es eine priorisierte Liste der Merkmale des Produktes oder der Dienstleistungen bereitstellt, welche das Projekt liefern soll. Die geforderten Merkmale werden als User Storys dargestellt. User Storys sind spezifische, das vorgeschlagene Produkt oder die Dienstleistung betreffende Anforderungen, die von verschiedenen Business Stakeholdern umrissen wurden. Jede User Story wird zugehörige User Story Abnahmekriterien haben (auch „Abnahmekriterien“ genannt), welche die Zielkomponenten sind, nach denen die Funktionalität der User Storys bewertet wird. Abnahmekriterien werden vom Product Owner nach seinem oder ihrem sachkundigen Verständnis der Kundenwünsche entwickelt. Der Product Owner kommuniziert dann die User Storys im Priorisierten Produkt Backlog an die Scrum Team-Mitglieder und erfragt deren Einverständnis. Abnahmekriterien sollten die Konditionen, welche die User Storys erfüllen müssen, deutlich darstellen. Klar definierte Abnahmekriterien sind ausschlaggebend für eine rechtzeitige und effektive Lieferung der in den User Storys definierten Funktionalitäten, was schlussendlich den Erfolg eines Projektes bestimmt.

Am Ende jedes Sprints überprüft der Product Owner die fertiggestellten zu liefernden Ergebnisse mithilfe dieser Kriterien und kann diese sowie die dazugehörigen User Storys individuell akzeptieren oder ablehnen. Wenn der Product Owner zu liefernde Ergebnisse akzeptiert, dann wird die User Story als Done angesehen. Eine klare Definition von Done ist entscheidend, weil es dabei hilft, Anforderungen zu verdeutlichen und es dem Team ermöglicht, den Qualitätsmaßstab einzuhalten. Des weiteren ermöglicht es dem Team, während der Arbeit mit den User Storys die Denkweise der Anwender zu übernehmen.

Abgelehnte User Storys werden wieder dem aktualisierten Priorisierten Produkt Backlog hinzugefügt, um während in zukünftigen Sprints vervollständigt zu werden. Die Ablehnung einiger individuell zu liefernden Ergebnisse und deren zugehöriger User Storys ist keine Ablehnung des Endprodukts oder Produkt-Inkremments. Das Produkt oder das Produkt-Inkrement könnte potenziell lieferbar sein, auch wenn ein paar User Storys abgelehnt werden.

Abbildung 5-1 stellt das Konzept der Abnahmekriterien zusammen mit dem Produkt-Inkrement im Flussdiagramm dar.

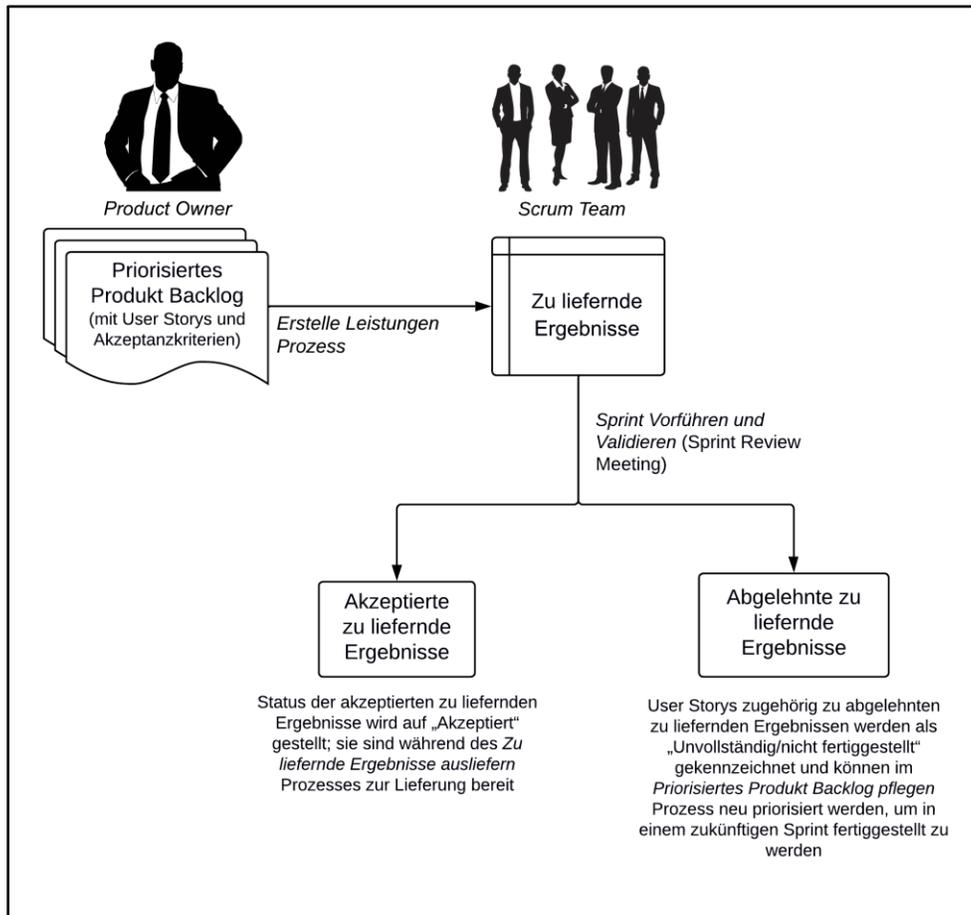


Abbildung 5-1: Projekt-Inkrement Flussdiagramm

5.4.1 Abnahmekriterien schreiben

Abnahmekriterien sind User Story spezifisch und kein Ersatz für eine Anforderungsliste.

Beispiel:

Persona: Janine ist eine verheiratete, 36jährige Berufstätige mit einer Familie und drei Kindern. Sie ist eine beschäftigte und erfolgreiche Frau, die Ihren Beruf und ihr Privatleben gut unter einen Hut bringt. Sie kommt mit moderner Technologie gut zurecht und ist eine der ersten, die neue Produkte und Dienste anwendet. Ueber verschiedener Geräte ist sie ständig im Netz und kauft regelmäßig auf E-Commerce-Portalen ein.

User Story: „Als online Kundin Janine sollte ich in der Lage sein, meinen Konzeptentwurf von jedem meiner Geräte abzuspeichern und anzusehen, so dass ich den Bestellprozess nach meinen Wünschen fertigstellen kann.“

Akzeptanz kriterium:

- Jede im Gange befindliche Bestellung des eingeloggten Benutzers muss alle fünf Sekunden als vorläufige Bestellung abgespeichert werden
- Neuer Auftragseingang muss als Mitteilung auf jedem Gerät erscheinen, mit dem sich der Benutzer anmeldet.

Für jeden Product Owner ist wichtig zu beachten, dass User Storys, die die meisten, aber nicht alle Abnahmekriterien erfüllen, nicht als Done akzeptiert werden können. Scrum Projekte werden in zeitlich begrenzten (time-boxed) Sprints durchgeführt, mit einem zugehörigem Sprint Backlog für jeden Sprint. Häufig ist der letzte Arbeitsschritt der schwierigste Teil einer Anwender Story und kann länger dauern als erwartet. Wenn unvollständige User Storys Teilanerkennung als Done bekämen und in den nächsten Sprint mit hinüber getragen würden, könnte der Fortschritt des folgenden Sprints gestört werden. Deshalb ist der Done Status entweder ja oder nein. Eine User Story kann nur Done (erledigt) oder not Done (nicht erledigt) sein.

5.4.2 Definition von Ready

Die Definition von Ready ist eine Reihe von Regeln oder Kriterien, die für jede User Story im priorisierten Product Backlog gelten. Eine User Story muss die Definition von Ready erfüllen, bevor sie zur Schätzung und Aufnahme in einen Sprint in Betracht gezogen wird. Die Definition of Ready legt die Verantwortung auf den Product Owner, die Anforderungen für jede User Story richtig zu definieren. Ohne korrekt definierte Anforderungen ist es unmöglich, verlässliche Abschätzungen zu erhalten, und das Scrum-Team ist möglicherweise nicht in der Lage, die erforderliche Projektarbeit effektiv zu erledigen.

Die Definition der Ready-Kriterien sollte vorzugsweise durch den Scrum Guidance Body definiert werden. Es kann jedoch auch eine projekt- oder organisationsspezifische Definition von Ready-Kriterien geben, die vom Product Owner hinzugefügt oder aktualisiert werden müssen. Es kann auch Ergänzungen oder Aktualisierungen der Definition von Ready durch das Scrum Team geben.

Das Scrum Team verpflichtet sich, an den User Storys zu arbeiten, die die Definition der Ready-Kriterien erfüllen. Die Überprüfung der Product Backlog Einträge anhand der Definition von Ready Kriterien ist eine kontinuierliche Aktivität, die als Teil des Prozesses Priorisiertes Produkt Backlog verfeinern stattfindet.

Einige Definition von Ready-Kriterien sind:

- Die User Storys sind so detailliert geschrieben, dass sie vom Scrum Team verstanden werden und zur Schätzung verwendet werden können.
- User Storys haben klar definierte Abnahmekriterien
- Alle zugehörige Dokumentation, die Klarheit über die User Storys liefert, ist enthalten
- Die User Storys klein genug, dass sie in einem einzigen Sprint abgeschlossen werden können.

5.4.3 Definition von Done (oder Done Kriterien)

Es gibt einen Hauptunterschied zwischen „Done-Kriterien“ und „Abnahmekriterien“. Während Abnahmekriterien spezifisch für individuelle User Storys sind, gelten Done-Kriterien als Regelsatz, der auf alle User Storys in einem festgelegten Sprint anzuwenden ist. Allgemeine Done-Kriterien können jede der folgenden Punkte einschließen:

- Von anderen Team-Mitgliedern überprüft
- Einheitentest der Anwender Story durchgeführt
- Qualitätssicherungstests durchgeführt
- Vollendung aller Anwender Story zugehörigen Dokumentation
- Alle Probleme sind gelöst
- Erfolgreiche Vorführung vor Business Stakeholdern/Geschäftsvertretern

Genau wie bei den Abnahmekriterien müssen alle Bedingungen der Done-Kriterien erfüllt werden, damit die User Story als Done angesehen werden kann. Das Scrum Team sollte eine Checkliste der allgemeinen Done-Kriterien benutzen, um sicherzustellen, dass eine Aufgabe abgeschlossen ist und die Ergebnisse der Definition von Done (DoD) gerecht werden. Eine klare Definition von Done ist entscheidend, denn sie hilft Unklarheiten zu beseitigen und ermöglicht dem Team, sich an den vorgegebenen Qualitätsmaßstab zu halten.

Die Definition von Done (oder die Done-Kriterien) wird normalerweise vom Scrum Guidance Body festgelegt und dokumentiert. Es kann jedoch projekt- oder organisationsspezifische Done Kriterien geben, die hinzugefügt oder aktualisiert werden müssen. Es kann auch Ergänzungen oder Anpassungen der Done Kriterien durch das Scrum Team geben.

Die zur Einhaltung der Dokumentationsanforderung des Projekts benötigten Daten und Datensätze können während der Durchführung der Sprints und Releases erzeugt werden. Review Meetings und das Schreiben von Design Dokumenten sind Maßnahmen, die (wenn einbezogen) dazu beitragen, die Einhaltung von internen und

externen Qualitätsstandards zu gewährleisten. Scrum Grundsätze wie kurze Durchläufe, Inkrements, Kundeneinbindung, die Adaptierung an sich ändernde Anforderungen und ständiges Anpassen des Umfangs, der Zeit und der Kosten innerhalb des Projektes gelten weiterhin.

5.4.4 Minimale Done Kriterien

Eine übergeordnete Geschäftseinheit mag verbindliche Minimale Done Kriterien verkünden, welche dann ein Teil der Abnahmekriterien für jede User Story dieser Geschäftseinheit werden. Jegliche von der Geschäftseinheit festgelegte Funktionalität muss diese Minimalen Done Kriterien erfüllen, um vom entsprechenden Product Owner akzeptiert zu werden. Die Einführung dieser Abnahmekriterien kann zu einer Kaskadierung von Abnahmekriterien für das Portfolio, das Programm und das Projekt führen (siehe Abbildung 5-2). Die allgemeinen Qualitätsstandards, Leitfäden und Vorlagen für das gesamte Portfolio werden vom Portfolio Product Owner festgelegt, während die Minimalen Done Kriterien auf dem Programm Level vom Programm Product Owner festgeschrieben werden. Somit werden die Abnahmekriterien einer User Story in einem Projekt vorbehaltlos alle Minimalen Done Kriterien übergeordneter Levels enthalten, soweit sie zutreffen.

Portfolio Product Owner	<ul style="list-style-type: none"> • Definiert die Minimalen Abnahmekriterien für das gesamte Portfolio • Überprüft zu liefernde Ergebnisse des Portfolios
Program Product Owner	<ul style="list-style-type: none"> • Legt die Minimalen Abnahmekriterien für das gesamte Programm fest, inklusive der Abnahmekriterien des entsprechenden Portfolios • Überprüft die zu liefernden Ergebnisse des Programms
Product Owner	<ul style="list-style-type: none"> • Legt die Minimalen Abnahmekriterien für das gesamte Projekt, inklusive der Abnahmekriterien des betreffenden Programms, fest • Überprüft die zu liefernden Ergebnisse des Projekts

Tabelle 5-1: Stufen der Abnahmekriterien

Falls minimale Done Kriterien definiert werden, können diese in die Dokumentation des Scrum Guidance Body aufgenommen werden, damit die Scrum Teams sich darauf beziehen können.

5.4.5 Akzeptieren oder Ablehnen eines Priorisierten Produkt Backlog Artikels

Zum Ende jeder Iteration beteiligen sich die betreffende Geschäftseinheit und die Business Stakeholdern an einem Sprint Review Meeting, in dem das Produkt-Inkrement dem Product Owner, Sponsor, Kunden und Anwendern vorgeführt wird. Von allen Business Stakeholdern wird Feedback gesammelt, doch nur der Product Owner hat die Berechtigung, nach den vereinbarten Abnahmekriterien zu beurteilen, ob eine bestimmte User Story als Done akzeptiert oder abgelehnt wird. Demnach sind die Abnahmekriterien entscheidend, um die Qualität aufrecht zu erhalten, und müssen vom Team eindeutig verstanden werden. Es liegt in der Verantwortung

des Scrum Masters sicher zu stellen, dass die Abnahmekriterien einer User Story nicht vom Product Owner in der Mitte eines Sprints geändert werden. Teilweise fertiggestellte Anwender Storys werden als not Done (nicht erledigt) abgelehnt und wieder in das Priorisierte Produkt Backlog eingespeist.

5.5 Qualitätsmanagement in Scrum

Der Kunde ist der wichtigste Business Stakeholdern in jedem Projekt. Deshalb ist es wichtig, die Bedürfnisse und Anforderungen des Kunden zu verstehen. Sowohl explizite als auch implizite Anforderungen des Kunden, die vor der Konzeption eines Produkts oder eines Service verstanden werden müssen, können als Stimme des Kunden (VOC) bezeichnet werden. Generell konzentriert sich der Product Owner in einem Scrum Umfeld auf die geschäftlichen Anforderungen und Ziele, welche zusammengefasst die Stimme des Kunden repräsentieren. Der Product Owner kann in großem Maße von der zur Verfügung gestellten Anleitung des Scrum Guidance Body profitieren (entweder durch Qualitätsdokumente, Standards oder durch Experten). Diese Spezialisten sollten mit dem Product Owner und dem Kunden zusammenarbeiten, um das geeignete Maß an Details und Informationen in der User Story zu gewährleisten, da User Storys die Grundlage des Erfolgs eines jeden Scrum Projekts sind.

Es sei zu beachten, dass externe Business Stakeholdern nicht direkt auf der Scrum Team Ebene beteiligt sind, sondern stattdessen primär mit dem Product Owner interagieren. In jedem Scrum Projekt kann der Kunde eines von beidem sein:

- Intern (das heißt Teil der Organisation)
- Extern (das heißt von außerhalb der Organisation)

Das Scrum Qualitätsmanagement ermöglicht es dem Kunden, jegliche Probleme frühzeitig zu erkennen und herauszufinden, ob das Projekt für ihn funktioniert oder nicht. In Scrum steht Qualität für Kundenzufriedenheit und ein funktionierendes Produkt, und nicht unbedingt dafür, willkürliche Metriken einzuhalten. Dieser ist ein entscheidender Unterschied aus Kundensicht, denn er investiert Zeit und Geld in das Projekt.

Scrum Qualitätsmanagement wird durch einen Drei-Stufen-Prozess vereinfacht:

1. Qualitätsplanung
2. Qualitätskontrolle
3. Qualitätssicherung

5.5.1 Qualitätsplanung

Eines der Leitprinzipien von Scrum ist, als höchste Priorität zuerst die Funktionalität zu entwickeln, die von höchster Priorität für den Kunden ist. Weniger wichtige Funktionen werden je nach Anforderungen des Kunden in nachfolgenden Sprints entwickelt oder komplett ausgelassen. Dieser Ansatz gibt dem Scrum Team die erforderliche Zeit, sich auf die Qualität der wesentlichen Funktionalität zu konzentrieren. Ein Hauptvorteil der Qualitätsplanung ist Reduzierung Technischer Schuld. Technische Schuld - auch Design-Schuld oder Code Schuld genannt - kann entstehen, während das Team an der Schaffung der primär zu liefernden Ergebnisse

des mit dem Projekt verbundenen (Software-) Produkts arbeitet. Es handelt sich dabei um den Teil der Arbeit, der vom Team als minderwichtig angesehen und gar nicht oder nur unvollständig ausgeführt wird. Technische Schuld sammelt sich an und muss irgendwann bezahlt werden.

Ursachen von Technischer Schuld können Folgendes beinhalten:

- Schnelle Lösungen und Erstellen von zu liefernden Ergebnissen, welche den Standards für Qualität, Sicherheit, langfristigen Architekturzielen und so weiter nicht nachkommen
- Unzureichendes oder unvollständiges Testen
- Ungeeignete oder unvollständige Dokumentation
- Fehlende Abstimmung zwischen den verschiedenen Team-Mitgliedern oder Scrum Teams, wenn Teams anfangen, abgesondert zu arbeiten, mit geringerem Fokus auf die letztendliche Integration der notwendigen Komponenten, um das Projekt oder Programm erfolgreich zu machen
- Mangelhafter Austausch von betriebswirtschaftlichen Kenntnissen und Prozess-Kenntnissen zwischen den Business Stakeholdern und den Projekt Teams
- Zu starker Fokus auf kurzfristige Projektziele anstatt auf langfristige Geschäftsziele. Ein solches Versehen kann zu mangelhafter Qualität der zu liefernden Ergebnisse führen, was erhebliche Kosten in der Instandhaltung und Verbesserung nach sich zieht.

In Scrum Projekten wird jegliche Technische Schuld nicht über einen Sprint hinaus getragen, da es klar definierte Akzeptanz- und Done-Kriterien gibt. Die Funktionalität muss diese Kriterien erfüllen, um als Done (erledigt) zu gelten. Bei der Pflege des Priorisierten Produkt Backlogs und der Priorisierung der Anwender Storys erzeugen die Teams regelmäßig funktionierende abzuliefernde Ergebnisse und verhindern so die Ansammlung erheblicher Technischer Schuld. Der Scrum Guidance Body kann ebenfalls Dokumentation und Definition von Prozessen enthalten, die zu einer Verminderung der Technischen Schuld verhelfen können. Um die Menge der Technischen Schuld minimal zu halten, ist es wichtig, das erwartete Produkt eines Sprints und des Projekts zu definieren, sowie die jeweiligen Abnahmekriterien, jegliche Entwicklungsmethoden, denen gefolgt werden muss, und die Hauptverantwortungen der Scrum Team-Mitglieder in Bezug auf Qualität.

Die Abnahmekriterien zu definieren ist ein wichtiger Teil der Qualitätsplanung und ermöglicht es während des Projekts, eine effektive Qualitätskontrolle durchzuführen. Technische Schuld kann bei traditionellen Projektmanagement Methoden eine echte Herausforderung darstellen, wenn Entwicklung, Überprüfung, Dokumentation usw. nacheinander und oftmals von verschiedenen Personen durchgeführt werden, ohne eine einzelne verantwortliche Person für ein bestimmtes zu lieferndes Arbeitsergebnis zu haben. Dies hat zur Folge, dass sich Technische Schuld anhäuft, was zu erheblich höheren Kosten in der Instandhaltung, Integration und dem Release in der Endphase des Projekts führt.

Außerdem sind die Änderungskosten unter solchen Umständen sehr hoch, da zu einem späteren Zeitpunkt des Projekts Probleme auftauchen werden. Der Scrum Rahmen verhindert solche Angelegenheiten in Verbindung mit Technischer Schuld, indem sichergestellt wird, dass Done (erledigte) zu liefernde Ergebnisse mit Abnahmekriterien als Teil des Sprint Backlogs definiert sind und Hauptaufgaben wie Entwicklung, Überprüfung und Dokumentation als Teil des selben Sprints vom selben Scrum Team erledigt werden.

5.5.1.1 Kontinuierliche Integration und nachhaltiges Arbeitstempo

Ein nachhaltiges Arbeitstempo zu halten ist einer der wichtigsten Grundsätze von Scrum. Ein nachhaltiges Arbeitstempo überträgt sich in gehobene Mitarbeiterzufriedenheit, Stabilität und verbesserte Genauigkeit der Schätzwerte, was schlussendlich zusammengefasst zu einer erhöhten Kundenzufriedenheit führt. Um ein wahrhaft qualitativ hochwertiges Produkt zu entwickeln und ein gesundes Arbeitsklima zu erhalten, ist es wichtig, regelmäßig Integrations-Aktivitäten durchzuführen, anstatt den Integrationsprozess bis zum Schluss zu verschieben. Um in häufigen Intervallen Mehrwert zu bieten, sollte das Team kontinuierlich die Funktionalitäten eines jeden Priorisierten Produkt Backlog Artikels in jedem Sprint mit Hilfe von Verfahren wie kontinuierlicher Integration und automatisierter Produktprüfung entwickeln, überprüfen und integrieren. Aus Sicht des Teams ist es außerdem wichtig, sicher zu stellen, dass der Arbeitsaufwand des gegenwärtigen Sprints dem Arbeitsaufwand des vorhergehenden Sprints ähnelt, um durchgehend ein ausgeglichenes Tempo zu erhalten. Dies hilft dem Team, Arbeitsspitzen zu vermeiden, um zu gewährleisten, dass sie immer das nötige Maß an Leistung bringen können um die benötigte Arbeit zu erledigen. Ein nachhaltiges Arbeitstempo zu halten ist auch einer der wichtigsten Grundsätze von anderen Agile Praktiken wie z.B. DevOps.

5

5.5.2 Qualitätskontrolle und Qualitätssicherung

Nicht nur das Produkt, auch die Prozesse erfordern Qualität. Qualitätssicherung bezieht sich auf die Evaluierung von Prozessen und die sie bestimmenden Standards. Qualitätsmanagement stellt in einem Projekt sicher, dass Qualitätssicherung weiter als wichtig erachtet wird. Qualitätssicherungsaktivitäten werden als Teil der Arbeit durchgeführt. Tatsächlich ist die Qualitätssicherung ein wichtiger Faktor in der Definition von Done (erledigt). Die abzuliefernden Ergebnisse sind unvollständig, wenn passende Maßnahmen der Qualitätssicherung nicht durchgeführt wurden. Oft kommt Qualitätssicherung während eines Sprint Review Meetings zutage.

Der Product Owner des betreffenden Projekts, Programms oder Portfolios kann die Qualitätssicherungsaktivitäten überwachen und auswerten, um sicherzustellen, dass jedes Team weiterhin mit den vorgegebenen Qualitätsstandards einig ist und ihnen nachkommt. End-to-end Qualitätssicherung kann während der letzten Prüfung des Produkt, einem Release oder einem Sprint in Angriff genommen werden. Es kann ein Vergleich der Menge an Problemen gegenüber der Menge an fertiggestellten User Storys angestellt werden. Die Produktkomponenten mit Mängeln können als Priorisierte Produkt Backlog Artikel eingebaut werden, um entweder vom Team oder einer einzelnen Person während gewisser Zeiten des Sprints bearbeitet zu werden, abhängig von der Anzahl an Mängeln.

Die Qualitätskontrolle bezieht sich auf die Durchführung der geplanten Qualitätsmaßnahmen, die vom Scrum Team bei der Erstellung von potenziell versandfähigen zu Liefernden Ergebnissen ausgeführt werden. Qualitätskontrolle beinhaltet auch, aus allen vervollständigten Aktivitäten zum Zweck der kontinuierlichen Verbesserung zu lernen. Innerhalb des funktionsübergreifenden Teams ist es wichtig, die benötigten Fähigkeiten zur Durchführung der Qualitätskontrollaktivitäten zu haben. Während des Sprint Retrospektions-Meetings besprechen die Team-Mitglieder ihre Erfahrungen. Diese Lehren dienen als Input zur kontinuierlichen Verbesserung und tragen zur Optimierung der anhaltenden Qualitätskontrolle bei.

Der Scrum Guidance Body kann Prozesse und Dokumente definieren, welche von den Scrum Teams während der Durchführung des Projekts eingesehen werden können, um einen einheitlichen Qualitätsmaßstab sicherzustellen, den alle Projekte innerhalb der Firma einhalten.

5.5.3 Plan-Do-Check-Act (PDCA) Zyklus

Der Plan-Do-Check-Act-Zyklus – auch bekannt als Deming- oder Shewhart-Zyklus – wurde von Dr. W. Edwards Deming, dem Vater der modernen Qualitätskontrolle, und Dr. Walter A. Shewhart entwickelt. Es folgen wichtige Punkte Demings Philosophie:

- Management Richtlinien definieren Qualität. Wenn das Management dazu in der Lage ist, ein förderliches Umfeld zu bieten und ihre Mitarbeiter zu motivieren, die Qualität kontinuierlich zu verbessern, wird jeder Mitarbeiter dazu imstande sein, einen Beitrag zu einem Produkt von überragender Qualität zu leisten. Demings “Theory of Profound Knowledge” enthält Vorschläge dazu, was das Management tun sollte, um ein Umfeld zu schaffen, in dem jeder Mitarbeiter einen maßgeblichen Beitrag zur Qualitätsverbesserung leisten kann.

Deming änderte sein Plan-Do-Check-Act später zu Plan-Do-Study-Act (PDSA) um, weil er der Meinung war, dass der Begriff „Study“ die Analyse gegenüber der einfachen Überprüfung („Check“) stärker hervorhob.

Sowohl Scrum als auch der Deming-/Shewhart-/PDCA-Zyklus verstehen sich als iterative Methoden, die sich auf kontinuierliche Verbesserungen konzentrieren. Abbildung 5-3 zeigt die Stufen des PDCA Zyklus auf und deren Zusammenhang mit diversen Scrum Prozessen.

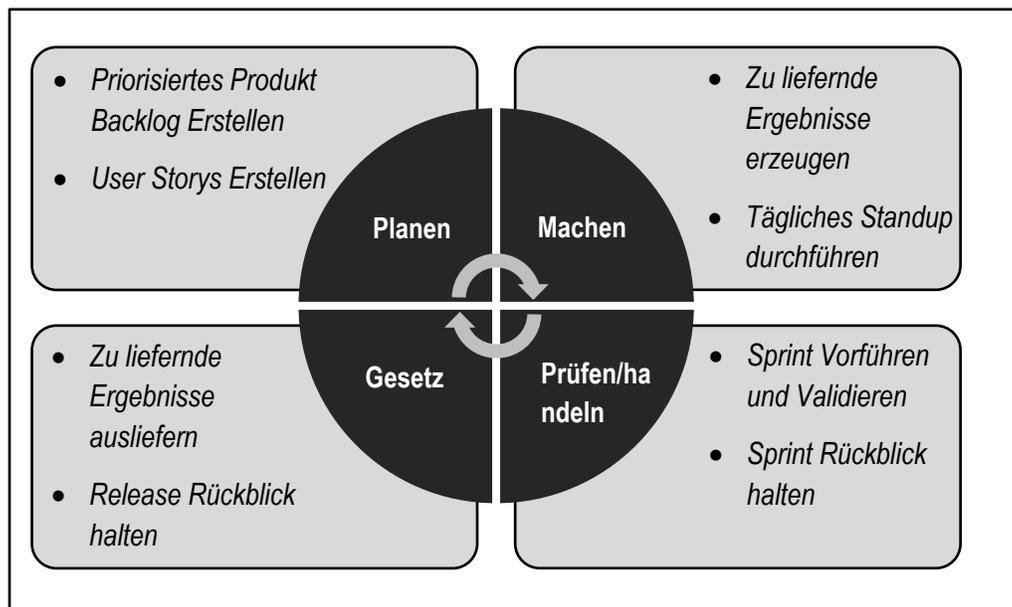


Abbildung 5-2: PDCA Zyklus in Scrum

5.6 Zusammenfassung der Verantwortlichkeiten

Kernrollen	Verantwortlichkeiten
Scrum Guidance Body	<ul style="list-style-type: none"> • Legt die Definition von Done fest • Stellt den Rahmen und Leitfaden für die Entwicklung von Abnahmekriterien • Definiert eine Reihe von Hilfsmitteln, die vom Scrum Team benutzt werden können, um das Produkt zu entwickeln und zu verifizieren
Portfolio Product Owner	<ul style="list-style-type: none"> • Definiert die Minimalen Abnahmekriterien für das gesamte Portfolio • Überprüft zu liefernde Ergebnisse des Portfolios
Portfolio Scrum Master	<ul style="list-style-type: none"> • Stellt sicher, dass ein nachhaltiges Arbeitstempo eingehalten wird, in dem der Fokus auf der Qualität der Features statt nur auf der Geschwindigkeit liegt
Programm Product Owner	<ul style="list-style-type: none"> • Legt die Minimalen Abnahmekriterien für das gesamte Programm fest inklusive der Abnahmekriterien des entsprechenden Portfolios • Überprüft der zu liefernde Ergebnisse des Programms
Programm Scrum Master	<ul style="list-style-type: none"> • Stellt sicher, dass ein nachhaltiges Arbeitstempo eingehalten wird, in dem der Fokus auf der Qualität der Features statt nur auf der Geschwindigkeit liegt
Business Stakeholdern	<ul style="list-style-type: none"> • Überprüft und liefert Feedback zu den zu liefernden Ergebnissen • Arbeiten eng mit dem Product Owner und dem Scrum Team zusammen
Product Owner/Chief Product Owner	<ul style="list-style-type: none"> • Legt die geschäftlichen Anforderungen an das Produkt fest und definiert Anforderungen unmissverständlich im Priorisierten Produkt Backlog • Bewertet die Realisierbarkeit und gewährleistet, dass zu liefernde Ergebnisse den Qualitätsansprüchen entsprechen • Legt die Minimalen Abnahmekriterien für das gesamte Projekt, inklusive der Abnahmekriterien des betreffenden Programms, fest • Fördert die Erstellung von Abnahmekriterien für User Storys • Überprüft und validiert die zu liefernden Ergebnisse während des <i>Sprint Vorführen und Validieren</i> Prozesses
Scrum Master/Chief Scrum Master	<ul style="list-style-type: none"> • Ermöglicht eine „Team zuerst“ Mentalität, wenn es um Qualität geht • Beseitigt Hindernisse im Umfeld, die die Qualität der zu liefernden Ergebnisse und Prozesse stören könnten • Stellt sicher, dass ein nachhaltiges Arbeitstempo eingehalten wird, in dem der Fokus auf der Qualität der Features, statt nur auf der Geschwindigkeit liegt • Stellt sicher, dass alle Team-Mitglieder, inklusive dem Product Owner, die Scrum Prozesse korrekt einhalten
Scrum Team	<ul style="list-style-type: none"> • Entwickelt und pflegt alle zu liefernden Ergebnisse während der Sprints bis zur Übergabe an den Endanwender • Praktiziert und fördert eine gute Kommunikation, damit die Anforderungen klar und vollständig verstanden werden • Teilt Wissen, um sicherzustellen, dass Team-Mitglieder mit den ganzen Features vertraut sind und damit von der Erfahrung anderer profitieren kann • Setzt angebrachte Änderungen an den zu liefernden Ergebnissen schnell um • Erfüllt die Done Kriterien für jedes zu Liefernde Ergebnis

Tabelle 5-2: Zusammenfassung der Verantwortlichkeiten hinsichtlich Qualität

5.7 Scrum versus traditionelles Projekt Management

Auch wenn es durchaus Gemeinsamkeiten in Scrum und traditionellen Projekt Management Methoden hinsichtlich der Definition von „Qualität“ gibt (d.h. die Fähigkeit des fertigen Produkts oder der fertigen abzuliefernden Ergebnisse, die Abnahmekriterien zu erfüllen und den vom Kunden erwarteten geschäftlichen Wert zu erzielen), gibt es Unterschiede im Hinblick darauf, was die Vorgehensweisen bzgl. der Implementierung und des Erreichens des geforderten Qualitätsmaßstabs angeht.

In traditionellen Projekt Management Methoden verdeutlichen die Anwender ihre Erwartungen; der Projekt Manager definiert diese Erwartungen in messbaren Bedingungen und erhält die Zustimmung der Anwender. Nach detaillierter Planung entwickelt das Projekt Team innerhalb eines abgesprochenen Zeitraums das Produkt. Falls irgendeine der abgesprochenen Kriterien geändert werden sollen, kann dies nur durch ein offizielles Changemanagementsystem geschehen, worin die Auswirkung der Änderung abgeschätzt wird und der Projekt Manager die Zustimmung aller betroffenen Business Stakeholdern bekommt.

In Scrum hingegen arbeitet der Product Owner mit dem Scrum Team zusammen und definiert die Abnahmekriterien für die Anwender Storys, die das zu liefernde Produkt betreffen. Das Scrum Team entwickelt das Produkt in einer Reihe von kurzen Iterationen, sogenannten Sprints.

Der Product Owner kann Änderungen an den Anforderungen vornehmen, um mit den Anwenderbedürfnissen mitzuhalten und diese Änderungen können vom Scrum Team aufgegriffen werden, entweder durch ein Beenden des derzeitigen Sprints oder durch ein Aufnehmen der geänderten Anforderungen im nächsten Sprint, da jeder Sprint nur von kurzer Dauer ist (d.h. ein bis sechs Wochen).

Einer der Hauptvorteile von Scrum ist der Schwerpunkt darauf, ein potenziell versandfähiges zu lieferndes Ergebnis am Ende jedes Sprint Zyklus zu entwickeln, statt nur am Ende des gesamten Projekts. Deshalb überprüfen, genehmigen und akzeptieren sowohl der Product Owner, als auch die Kunden kontinuierlich nach jedem Sprint die zu liefernden Ergebnisse. Auch wenn ein Scrum Projekt vorzeitig beendet wird, so ist dennoch ein Mehrwert durch die entwickelten zu liefernden Ergebnisse aus den einzelnen Sprints vorhanden.

6. Änderungen

6.1 Einleitung

Jedes Projekt ist Änderungen unterworfen, egal welche Methode oder welcher Rahmen benutzt wird. Für Mitglieder von Projektteams ist es dringend notwendig zu verstehen, dass der Scrum Entwicklungsprozess entworfen wurde, um Änderungen einfach einzubeziehen. Organisationen sollten versuchen die Vorteile, die Änderungen mit sich bringen, zu maximieren und jegliche negativen Auswirkungen durch sorgfältige Änderungsmanagement-Prozesse in Übereinstimmung mit den Scrum Grundsätzen zu minimieren.

Change, wie im *Scrum Body of Knowledge (SBOK®)* definiert, ist auf Folgendes anwendbar:

- Portfolios, Programme und/oder Projekte in jeder Branche
- Produkte, Dienstleistungen oder jegliche an Business Stakeholdern zu liefernde Ergebnisse
- Projekte jeglicher Größe oder Komplexität

Der Begriff "Produkt" in dieser Wissenssammlung kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann auf jedes Projekt in jeder Branche effektiv angewendet werden — von kleinen Projekten oder Teams mit lediglich sechs Team-Mitgliedern bis hin zu großen, komplexen Projekten mit mehreren Hundert Team-Mitgliedern.

Dieses Kapitel ist in folgende Abschnitte aufgeteilt:

6.2 Leitfaden für die Kernrollen—Dieser Teil klärt die Frage, welche Abschnitte für jede Scrum Rolle relevant sind: Product Owner, Scrum Master und Scrum Team.

6.3 Überblick—Dieser Abschnitt definiert das Konzept der Änderung, speziell im Zusammenhang mit Scrum Prozessen. Es wird außerdem erklärt, wie mit Change Requests (Änderungsanfragen) in Scrum Prozessen umgegangen wird.

6.4 Änderungen in Scrum—Dieser Abschnitt erklärt, wie wichtig es ist, Änderungen in einem Scrum Projekt effektiv zu handhaben. Weiterhin befasst es sich damit, wie Flexibilität und Stabilität durch angemessene Handhabung der Change Requests während des gesamten Projekts erreicht werden können.

6.5 Änderungen integrieren—Dieser Abschnitt gibt Aufschluss darüber, wie Change Requests unter Anwendung des Scrum Rahmens bewertet und akzeptiert (oder abgelehnt) werden.

6.6 Änderungen auf der Programm- und Portfolio-Ebene—Dieser Abschnitt beschreibt die Auswirkungen von Änderungen auf der Programm- und Portfolio-Ebene.

6.7 Zusammenfassung der Verantwortlichkeiten—Dieser Abschnitt definiert die Verantwortlichkeiten der Projekt Team-Mitglieder im Zusammenhang mit Änderungsmanagement.

6.8 Scrum versus traditionelles Projekt Management— Dieser Abschnitt hebt die Vorzüge des Änderungsmanagements nach der Scrum Methode gegenüber den traditionellen Projekt Management Modellen hervor.

6.2 Leitfaden für die Kernrollen

1. Product Owner — Änderungen innerhalb eines Projekts einzuleiten liegt primär in der Verantwortung des Product Owners; deshalb ist das gesamte Kapitel maßgebend für diese Kernrolle.
2. Scrum Master — Der Scrum Master sollte mit diesem kompletten Kapitel vertraut sein, besonders mit den Abschnitten 6.3, 6.4, 6.5 und 6.7.
3. Scrum Team — Das Scrum Team sollte sich hauptsächlich auf die Abschnitte 6.3, 6.4.2, 6.5, und 6.5 konzentrieren.

6.3 Überblick

In allen Projekten sind Änderungen unumgänglich. In der heutigen konkurrenzbetonten Welt, in der Technologie, Marktverhältnisse und Geschäftsstrukturen im ständigen Wandel sind, ist Veränderung die einzige Konstante.

Einer der Scrum Hauptgrundsätze ist die Anerkennung, dass a) Business Stakeholdern (z.B. Kunden, Anwender und Sponsoren) ihre Meinung bezüglich ihrer Wünsche und Bedürfnisse innerhalb eines Projekts ändern (was manchmal als „requirements churn“ bezeichnet wird) und b) dass es sehr schwer, wenn nicht sogar unmöglich für die Business Stakeholdern ist, die gesamten Anforderungen zu Beginn des Projekts zu definieren.

Scrum Entwicklungsprojekte greifen Änderungen positiv auf, indem sie kurze Entwicklungszyklen benutzen, um nach jedem Sprint Kundenfeedback in die zu liefernden Ergebnisse einzubauen. Dies ermöglicht dem Kunden, regelmäßig mit den Scrum Team-Mitgliedern zu interagieren, Produktinkremente bei Fertigstellung zu sehen und Anforderungen früher in der Entwicklungsphase zu ändern. Auch die Portfolio oder Programm Management Teams können auf ihrem Level auf Change Requests hinsichtlich Scrum Projekten reagieren.

Scrum verkörpert ein Grundprinzip des Agile Manifesto (Fowler und Highsmith, 2001): „Auf Änderung zu reagieren ist wichtiger als einem Plan zu folgen.“ Scrum folgt der Grundlage, Änderungen anzunehmen und in einen Vorteil der Konkurrenz gegenüber umzuwandeln. Deshalb ist es wichtiger, flexibel zu sein, als einem strikten vordefinierten Plan zu folgen. Dies bedeutet, es ist entscheidend einen anpassungsfähigen Ansatz des Projekt-Managements zu haben, der kurzfristige Änderungen während der ganzen Produkt- oder Dienstleistungsentwicklung möglich macht.

Sich Veränderungen anpassen zu können ist ein Hauptvorteil des Scrum Rahmens. Auch wenn es für Projekte jeder Branche geeignet ist, kann Scrum besonders effektiv sein, wenn Produkte oder andere Projektanforderungen nicht vollständig verstanden werden oder im Voraus nicht gut definiert werden können, wenn der Produktmarkt unbeständig ist und/oder der Schwerpunkt darauf liegt, das Team flexibel genug zu

machen, dass mit veränderten Anforderungen umgehen kann. Speziell für komplexe Projekte mit einer Menge an Unsicherheiten ist Scrum nützlich. Langfristige Planung und Prognosen sind normalerweise für solche Projekte ineffektiv und beinhalten eine Menge Risiken. Scrum leitet das Team durch *Transparenz*, *Inspektion*, und *Adaptierung* zum bestmöglichen Geschäftsergebnis.

6.3.1 Nicht genehmigte und genehmigte Change Requests

Änderungsanfragen werden üblicherweise als Change Requests eingereicht. Change Requests bleiben solange nicht genehmigt, bis sie formal genehmigt werden. Der Scrum Guidance Body legt normalerweise einen Prozess zum Genehmigen und Verwalten von Änderungen für die gesamte Organisation fest. Ist kein offizieller Prozess bestimmt worden, wird empfohlen, dass kleinere Änderungen, die keinen erheblichen Einfluss auf das Projekt haben, direkt vom Product Owner genehmigt werden. Die Toleranz solch kleiner Änderungen kann auf organisatorischer Ebene oder vom Sponsor des jeweiligen Projekts definiert werden. In den meisten Projekten können 90% der Change Requests als kleine Änderungen eingestuft werden, die vom Product Owner genehmigt werden sollten. Somit spielt der Product Owner eine sehr wichtige Rolle im Änderungsmanagement eines Scrum Projekts. Änderungen, die über die Obergrenze für die Genehmigung durch den Product Owner hinausgehen, bedürfen möglicherweise der gemeinsamen Genehmigung durch relevante Business Stakeholdern und den Product Owner.

Änderungen, die über den Toleranzlevel des Product Owner hinausgehen, brauchen eventuell die Genehmigung der betreffenden Business Stakeholdern, welche mit dem Product Owner zusammenarbeiten.

Manchmal, wenn eine Änderungsanfrage erhebliche Auswirkungen auf das Projekt oder die Organisation haben könnte, wird die Genehmigung vom Senior Management (z.B. Executive Sponsor, Portfolio Product Owner, Programm Product Owner oder Chief Product Owner) benötigt.

Das Projekt betreffende Change Requests werden während der Prozesse *Epics entwickeln*, *Priorisiertes Produkt Backlog erstellen* und *Priorisiertes Produkt Backlog verfeinern* besprochen und genehmigt. *Genehmigte Change Requests* werden dann zusammen mit den anderen Produktanforderungen und ihren zugehörigen User Storys priorisiert und in das Priorisierte Produkt Backlog eingearbeitet.

Abbildung 6-1 fasst den Änderungsanfrage-Prozess zusammen und Abbildung 6-2 erklärt, wie das Priorisierte Produkt Backlog mit den genehmigten Änderungen aktualisiert wird.

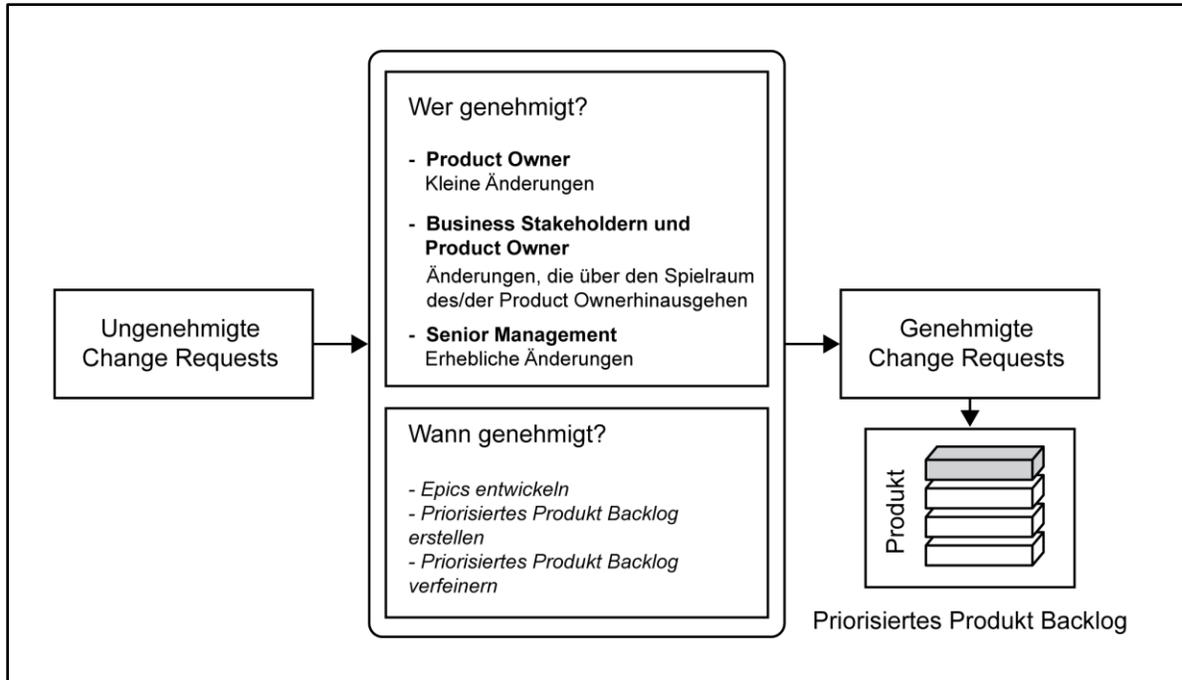


Abbildung 6-1: Beispiel genehmigter Änderungsprozess

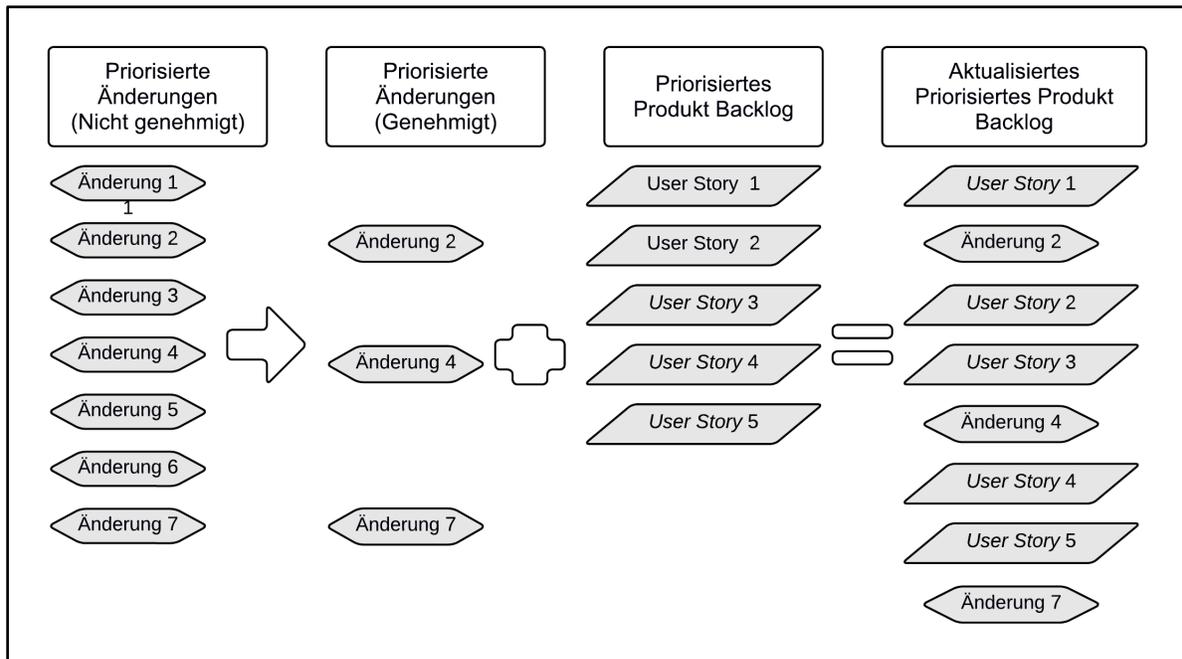


Abbildung 6-2: Aktualisieren des Priorisierten Produkt Backlogs mit genehmigten Änderungen

6.4 Änderungen in Scrum

6.4.1 Das Gleichgewicht an Flexibilität und Stabilität halten

Scrum verhilft Organisationen flexibler und offener für Veränderungen zu werden. Trotzdem ist es wichtig zu verstehen, dass, obwohl der Scrum Rahmen Wert auf Flexibilität legt, es auch wichtig ist, die Stabilität während des Änderungsprozesses zu wahren. Extreme Flexibilität ist genauso unproduktiv, wie extreme Starrheit ineffektiv ist. Der Schlüssel ist die richtige Ausgeglichenheit zwischen Flexibilität und Stabilität zu finden, um die Arbeit fertigzustellen. Deshalb benutzt Scrum die schrittweise iterative Lieferung und seine anderen Merkmale und Grundsätze, um diesen Ausgleich zu erreichen. Scrum erreicht Flexibilität, indem Change Requests während des Projekts jederzeit gestellt und genehmigt werden können; dennoch werden sie erst priorisiert, wenn das Priorisierte Produkt Backlog erschaffen oder aktualisiert wird. Gleichzeitig garantiert Scrum Stabilität durch ein festgesetztes Sprint Backlog und durch Fernhaltung von Beeinflussungen auf das Scrum Teams während eines Sprints.

In Scrum sind während eines Sprints alle Anforderung bezüglich des laufenden Sprints fixiert. Bis zum Ende des Sprints werden keine Änderungen einbezogen, es sei denn, die Änderung wird als bedeutend genug angesehen, um den Sprint zu stoppen. Ist die Änderung dringend erforderlich, wird der Sprint beendet und das Team trifft sich, um einen neuen Sprint zu planen. Auf diese Weise akzeptiert Scrum Änderungen, ohne das Release Datum ändern zu müssen.

6.4.2 Flexibilität erreichen

Aufgrund des iterativen Charakters von Scrum sowie der Konzepte von Transparenz, Inspektion und Anpassung in der empirischen Prozesssteuerung ist Flexibilität ein wesentlicher Bestandteil jeder Scrum-Implementierung. Scrum bietet einen anpassungsfähigen Mechanismus zum Projektmanagement, bei dem eine Änderung der Anforderungen aufgenommen werden kann, ohne den gesamten Projektfortschritt wesentlich zu beeinflussen. Es ist notwendig, sich im Entwicklungszyklus den entstehenden Geschäftsrealitäten anzupassen. Scrum erzielt Flexibilität durch fünf Hauptmerkmale (siehe Abbildung 6-3): iterative Produktentwicklung, Time-Boxing, funktionsübergreifende Teams, kundenwertorientierte Priorisierung und kontinuierliche Integration.

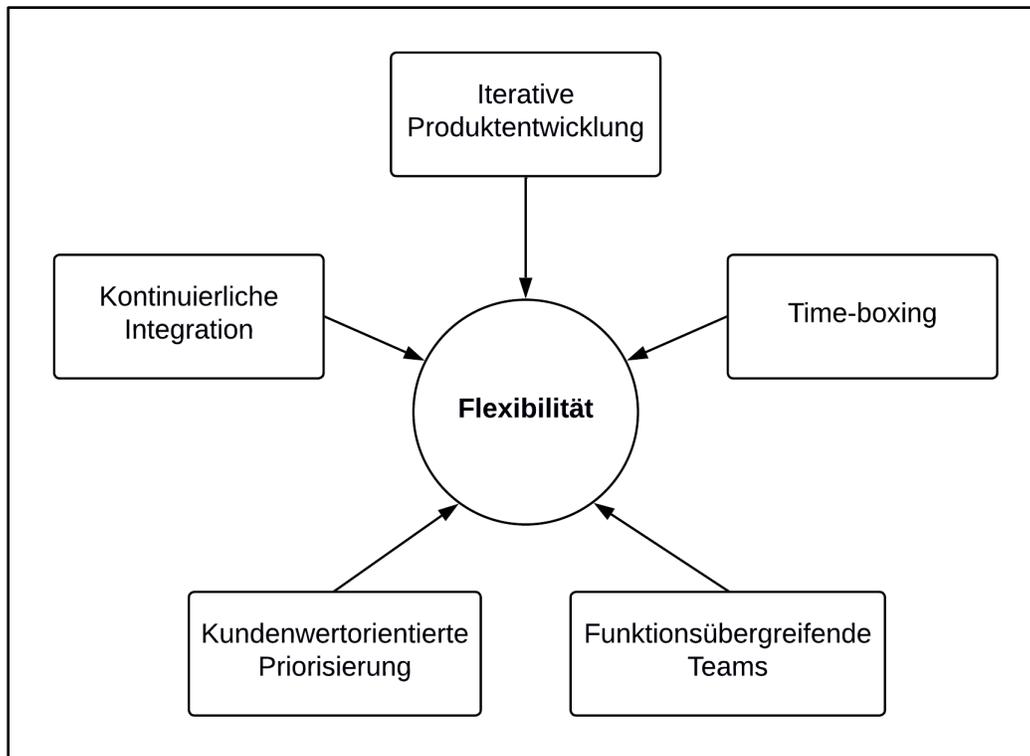


Abbildung 6-3: Scrum Merkmale um Flexibilität zu erreichen

6.4.2.1 Flexibilität durch iterative Produktentwicklung

Scrum verfolgt einen iterativen und inkrementellen Ansatz für die Entwicklung von Produkten und Dienstleistungen, der es ermöglicht, Änderungen nach jeder Stufe des Entwicklungsprozesses einzuarbeiten. Während der Entwicklung des Produkts kann ein Change Request von verschiedenen Seiten kommen, wie folgt:

1. Business Stakeholdern

Business Stakeholdern—insbesondere Sponsoren, Kunden und Anwender—können zu jeder Zeit des Projekts einen Change Request einreichen. Change Requests können aufgrund von Änderungen der Marktverhältnisse, organisatorischer Ausrichtung, gesetzlichen oder behördlichen Angelegenheiten oder verschiedenen anderen Ursachen entstehen. Zudem können Business Stakeholdern Change Requests einreichen, während sie die zu liefernden Ergebnisse der Prozesse *Sprint Vorführen und Validieren*, *Sprint Rückblick halten*, oder *Release Rückblick halten* überprüfen. Alle Change Requests werden dem Priorisierten Produkt Backlog hinzugefügt, sobald sie genehmigt wurden.

Abbildung 6-4 veranschaulicht einige Gründe für Business Stakeholdern, den Change Request Prozess einzuleiten.

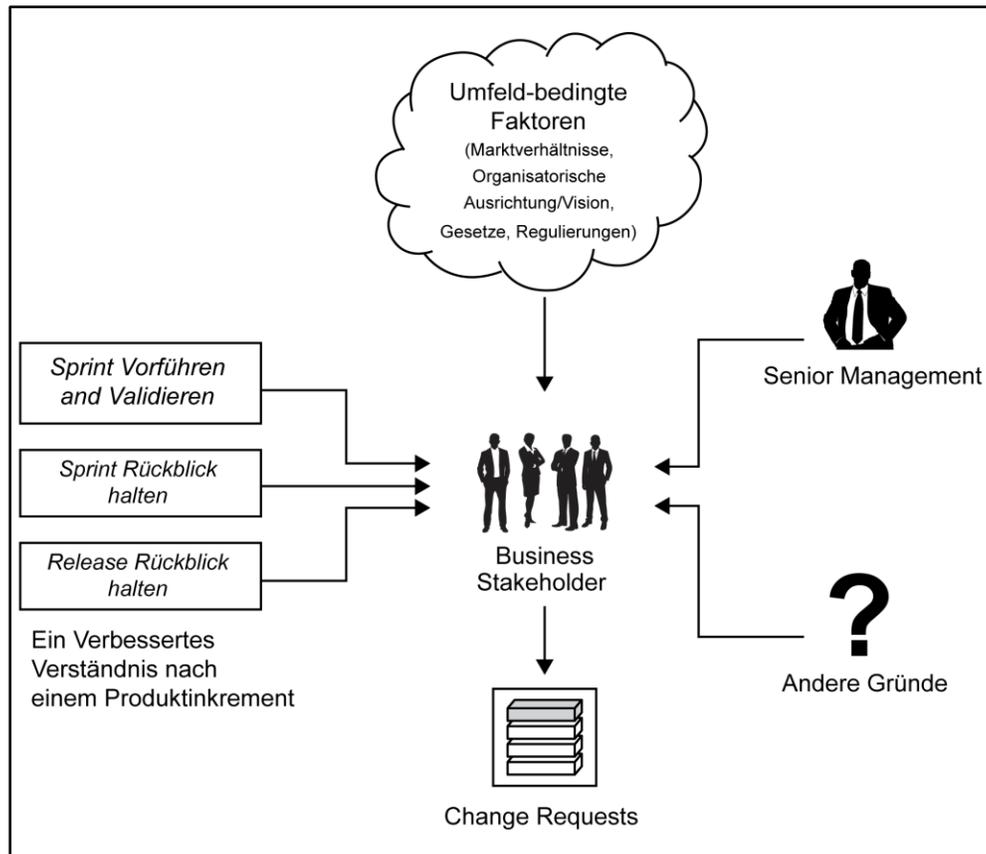


Abbildung 6-4: Beweggründe der Business Stakeholdern, Änderungen zu verlangen

2. Scrum Kernteam

Das Scrum Kernteam (d.h. der Product Owner, Scrum Master und Scrum Team) sind alle an der Entwicklung der zu liefernden Ergebnisse beteiligt. Anhaltende Interaktionen zwischen den Kernteam-Mitgliedern eines Scrum Teams und anderen, wie zum Beispiel anderen Scrum Teams des Projekts, internen und externen Projekt Business Stakeholdern, kann Scrum Kernteam-Mitglieder dazu motivieren, Änderungen oder Verbesserungen am Produkt, an der Dienstleistung oder einem anderen Teil des Projekts vorzuschlagen. Normalerweise werden solche Änderungen—wie alle anderen auch—als Change Requests festgehalten und der Product Owner trifft eine endgültige Entscheidung darüber, ob die vom Scrum Team und Scrum Master vorgeschlagenen Änderungen als offizielle Change Requests gekennzeichnet werden sollen.

Zeitweise kann es sein, dass Schwierigkeiten bei der Erstellung einzelner zu liefernder Ergebnisse auftauchen, welche Change Requests zur Folge haben können. Zum Beispiel entscheidet das Team sich vielleicht dazu, ein neues Feature hinzuzufügen oder zu modifizieren, um das Gesamtprodukt zu verbessern.

In den meisten Scrum Projekten werden Änderungsvorschläge des Scrum Kernteams während ihrer Arbeit an *Zu liefernde Ergebnisse erzeugen* oder ihrer Teilnahme am *Tägliche Standup Meeting* oder *Sprint Retrospektions-Meetings* erzeugt. Abbildung 6-5 veranschaulicht einige Gründe, warum das Scrum Kernteam Change Requests einreichen mag.

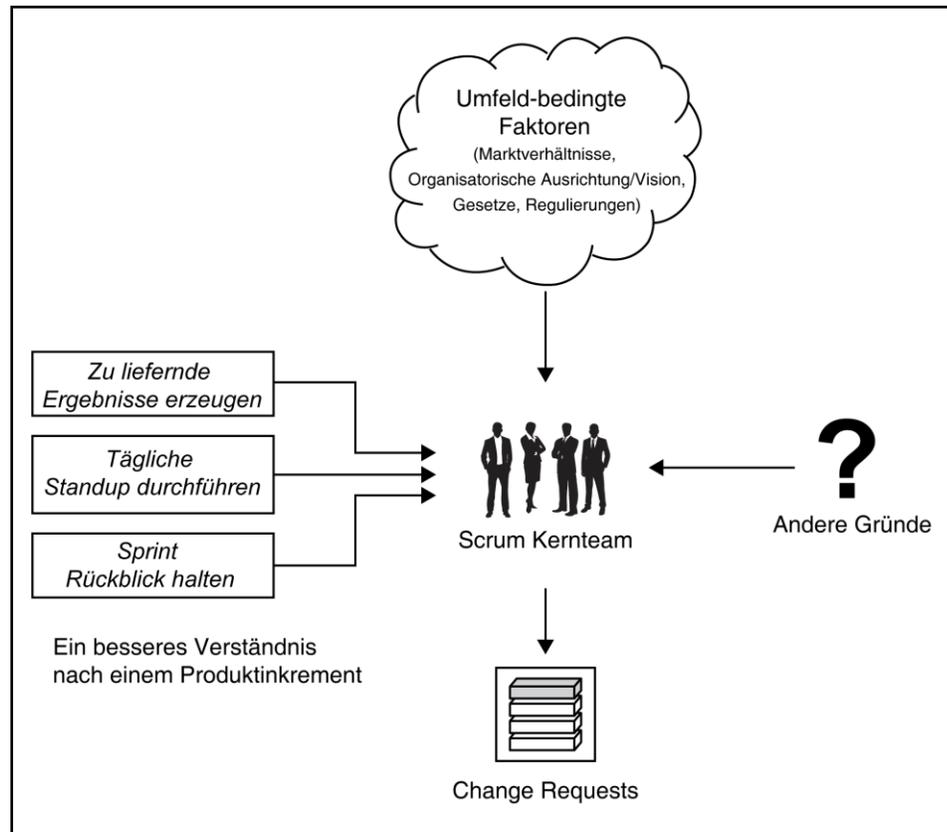


Abbildung 6-5: Beweggründe des Scrum Kernteams, Änderungen anzufordern

3. Senior Management

Das Senior Management—Portfolio und Programm Management eingeschlossen—kann Änderungen vorschlagen, die das Projekt beeinflussen. Als Gründe können Änderungen in der strategischen Ausrichtung des Unternehmens, dem Wettbewerbsumfeld, finanzierungsbezogene Angelegenheiten und vieles mehr genannt werden. Beachten Sie, dass solche Änderungen dem Priorisierten Produkt Backlog hinzugefügt werden und dann den normalen Änderungsmanagement Prozess durchlaufen. Falls eine dieser Änderungen dringlich ist, kann es notwendig sein, den betroffenen Sprint vorzeitig zu beenden (siehe Abschnitt 6.6 für mehr Informationen).

4. Scrum Guidance Body

Der Scrum Guidance Body kann alle Projekte beeinflussende Change Requests wegen jeder der folgenden Begebenheiten einreichen:

- Eine Veränderung der staatlichen Regulierungen (z.B. Datenschutz, Sicherheitsstandards oder eine neue Gesetzgebung)
- Anweisungen für Qualitäts-, Sicherheits- oder andere organisatorische Initiativen, die im ganzen Unternehmen eingeführt werden müssen.
- Bezugswerte oder bewährte Praktiken, um einen bestimmten Standard erfüllen
- erlerntes Wissen aus früheren Projekten zur Anwendung durch andere Scrum Teams

Das Scrum Markenzeichen ist, Änderungen gegenüber tolerant und anpassungsfähig zu sein. Scrum propagiert nicht, auf lange Sicht im Voraus feste Pläne zu schmieden, denn es arbeitet auf der Basis, dass die Projektentwicklung extremen Veränderungen und Risiken ausgesetzt ist. Das Ergebnis ist ein hoher Grad an Flexibilität und Toleranz Veränderungen gegenüber. Das Projekt wird inkrementell geplant, ausgeführt und verwaltet, so dass normalerweise Änderungen während des gesamten Projekts leicht eingearbeitet werden können.

6

6.4.2.2 Flexibilität durch Time-Boxing

Time-Boxing beschreibt die Einteilung in kurze Zeitfenster (time-boxes), binnen derer Arbeiten fertiggestellt werden müssen. Nach Ablauf des Zeitfensters (der Time-Box) werden nicht fertiggestellte Arbeiten in eine spätere Time-Box verschoben. Beispiele von Time-Boxing beinhalten, die täglichen Standup Meetings auf 15 Minuten zu limitieren und die Sprint Dauer auf ein bis vier Wochen festzusetzen.

Dass Sprints time-boxed sind trägt einen großen Teil dazu bei, Abgabetermine einzuhalten und eine höhere Produktivität zu erreichen. Fehler oder Probleme können frühzeitig erkannt und schnell behoben werden, was Flexibilität in Scrum-Projekten ermöglicht. Sprints fördern Ordnung und Beständigkeit in einem unbeständigen Arbeitsumfeld. Sie bieten eine Bühne zur Ergebnisabmessung und um schnell Feedback zu bekommen. Sprints erlauben auch regelmäßige Bewertungen des Fortschritts und der Methoden zur Verwaltung des Projekts, einschließlich des effektiven Änderungsmanagements. Die Selbstorganisation stellt sicher, dass die Mitglieder des Scrum-Teams die Flexibilität haben, alle Aufgaben zu bestimmen, an denen sie in einem Sprint arbeiten werden, und auch zu bestimmen, wie sie die Arbeit erledigen werden. Außerdem sorgt sie dafür, dass Teams von selbst motiviert sind, ihre selbst zugewiesenen Aufgaben zu erledigen, Engpässe zu beseitigen und den Wissensaustausch mit anderen Teammitgliedern zu fördern.

Durch die Benutzung von Time-Boxing in Sprints greift das Team regelmäßig den Kalkulationsprozess der vor ihm liegenden Arbeit auf, so dass die Prognose der benötigten Zeit und des Aufwands mit jedem nachfolgenden Sprint im Projektverlauf genauer wird. Diese iterativen Zyklen motivieren die Team-Mitglieder, geplante Ziele zu erreichen und stufenweise zum großen Ziel voranzuschreiten.

6.4.2.3 Flexibilität durch funktionsübergreifende und selbstorganisierte Teams

Die Selbstorganisation stellt sicher, dass die Mitglieder des Scrum-Teams die Flexibilität haben, alle Aufgaben zu bestimmen, an denen sie in einem Sprint arbeiten werden, und auch zu bestimmen, wie sie die Arbeit erledigen werden. Außerdem sorgt sie dafür, dass Teams von selbst motiviert sind, ihre selbst zugewiesenen Aufgaben zu erledigen, Engpässe zu beseitigen und den Wissensaustausch mit anderen Teammitgliedern zu fördern. Die funktionsübergreifende und selbstorganisierte Struktur der Scrum Teams ermöglicht Team-Mitgliedern, stark auf die gewünschten Sprint Ergebnisse fokussiert zu sein. Das Team hat für jeden Sprint festgesetzte Ziele und ist flexibel genug, vor Beginn des nächsten Sprints Änderungen an Zielsetzungen zu berücksichtigen.

Funktionsübergreifende Teams stellen sicher, dass alle erforderlichen Fähigkeiten und das benötigte Fachwissen im Team vorhanden sind, um die Arbeit des Projekts durchzuführen. Dieses bietet ein effizientes Arbeitsmodell, welches zur Erstellung zu liefernder Ergebnisse, die potenziell versandfähig und bereit zur Vorführung vor Product Ownern und/oder anderen Business Stakeholdern sind.

Die Selbstorganisation stellt sicher, dass Scrum Team-Mitglieder selbst festlegen *wie* sie die Arbeit an dem Projekt erledigen, ohne dass ein Senior Manager ihre Aufgaben in jedem Detail überwacht.

Funktionsübergreifende und selbstorganisierte Teams ermöglichen es der Gruppe anpassungsfähig zu sein, ihre Arbeit effektiv zu erledigen, kleinere Angelegenheiten oder Änderungen ohne Unterstützung oder Fachkenntnisse von außerhalb zu bewältigen und dabei erzeugen sie zu liefernde Ergebnisse die gfls. ausgeliefert werden können.

6.4.2.4 Flexibilität durch kundenwertorientierte Priorisierung

Die Priorisierung der Anforderungen und Arbeit an einem Scrum Projekt wird immer an dem Wert festgelegt, der dem Kunden geliefert wird. Zu Beginn eines Projekts werden zuerst die anfänglichen Anforderungen nach dem Wert den jede Anforderung bietet priorisiert—dies wird im Priorisierten Produkt Backlog dokumentiert. Während des *Priorisiertes Produkt Backlog verfeinern* Prozesses werden Anfragen bezüglich neuer Anforderungen oder Änderungen der Anforderungen bewertet. Falls die Änderung mehr Wert als andere bestehende Anforderungen bietet, wird sie entsprechend priorisiert und dem aktualisierten Priorisierten Produkt Backlog hinzugefügt. Somit bietet das Priorisierte Produkt Backlog das Mittel Änderungen einzubauen und wenn nötig neue Anforderungen hinzuzufügen.

Es gilt zu beachten, dass neue Anforderungen und zugefügte Änderungen die Prioritäten vorhandener User Storys im Priorisierten Produkt Backlog mindern können: Niedriger priorisierte User Storys könnten deshalb, abhängig von ihrer neuen Priorisierung, erst später umgesetzt werden. Da der Kunde sehr in der Priorisierung der Anforderungen und deren entsprechenden User Storys im Priorisierten Produkt Backlog eingebunden ist, ist sichergestellt, dass die Anforderungen, die der Kunde als „sehr wertvoll“ erachtet, früher fertiggestellt werden und das Projekt schnell signifikanten Mehrwert liefert.

6.4.2.5 Flexibilität durch kontinuierliche Integration

Indem sie kontinuierliche Integrationstechniken benutzen, können Scrum Team-Mitglieder, wann immer es möglich ist, neue und modifizierte Features in die zu liefernden Ergebnisse einarbeiten. Dies vermindert das Risiko, das dadurch entsteht, dass mehrere Team-Mitglieder Änderungen an redundanten Elementen vornehmen (z.B. überholter Code in Softwareprodukten, alte Designs für Herstellungsteile). Dies garantiert, dass nur die neusten Features oder Versionen bearbeitet werden und verhindert Kompatibilitätsprobleme.

6.5 Änderungen integrieren

Abhängig von der Branche und der Art des Projekts kann es sein, dass die Prioritäten der Features und Anforderungen an ein Projekt für einen längeren Zeitraum unverändert bleiben oder häufiger geändert werden. Wenn Projektanforderungen im Allgemeinen stabil sind, werden normalerweise während der Entwicklung nur kleine Änderungen am Priorisierten Produkt Backlog vorgenommen und Scrum Teams können sequenziell die Anforderungen abarbeiten, die den maximalen Kundenwert bieten werden, entsprechend der Priorität im Priorisierten Produkt Backlog. In solch einem stabilen Umfeld ist die Sprint-Länge normalerweise länger, von 4 bis 6 Wochen.

Dieselben Methoden bleiben effektiv, wenn sich Projektanforderungen während des gesamten Projekts ändern, zum Beispiel infolge von geänderten Geschäftlichen Anforderungen. Bevor der Sprint anfängt—während der Prozesse *Priorisiertes Produkt Backlog erstellen* oder *Priorisiertes Produkt Backlog verfeinern*—werden normalerweise die Anforderungen mit der höchsten Priorität aus dem Priorisierten Produkt Backlog ausgewählt, um in diesem Sprint fertiggestellt zu werden. Weil Änderungen im Priorisierten Produkt Backlog berücksichtigt wurden, braucht das Team nur zu entscheiden, wie viele Aufgaben es in einem Sprint in der verfügbaren Zeit und mit den bereitgestellten Ressourcen bewältigen kann. Das Änderungsmanagement wird im fortlaufenden Prozess der Priorisierung und dem Hinzufügen von User Storys zum Priorisierten Produkt Backlog ausgeführt.

6.5.1 Änderungen an einem Sprint

Wenn ein Change Request einen wesentlichen Einfluss auf einen laufenden Sprint hat, entscheidet der Product Owner nach Absprache mit den betreffenden Business Stakeholdern, ob die Änderung bis zum nächsten Sprint warten kann oder so dringend ist, dass der laufende Sprint beendet und ein neuer gestartet werden muss.

Der Scrum Rahmen legt klar dar, dass der Umfang eines Sprints nach seinem Anfang nicht mehr geändert werden kann. Wenn die geforderte Änderung so wichtig ist, dass das Ergebnis des Sprints ohne sie wertlos wäre, sollte der Sprint beendet werden. Wenn nicht, sollte die Änderung in einen späteren Sprint eingearbeitet werden (wie in Abbildung 6-6 gezeigt).

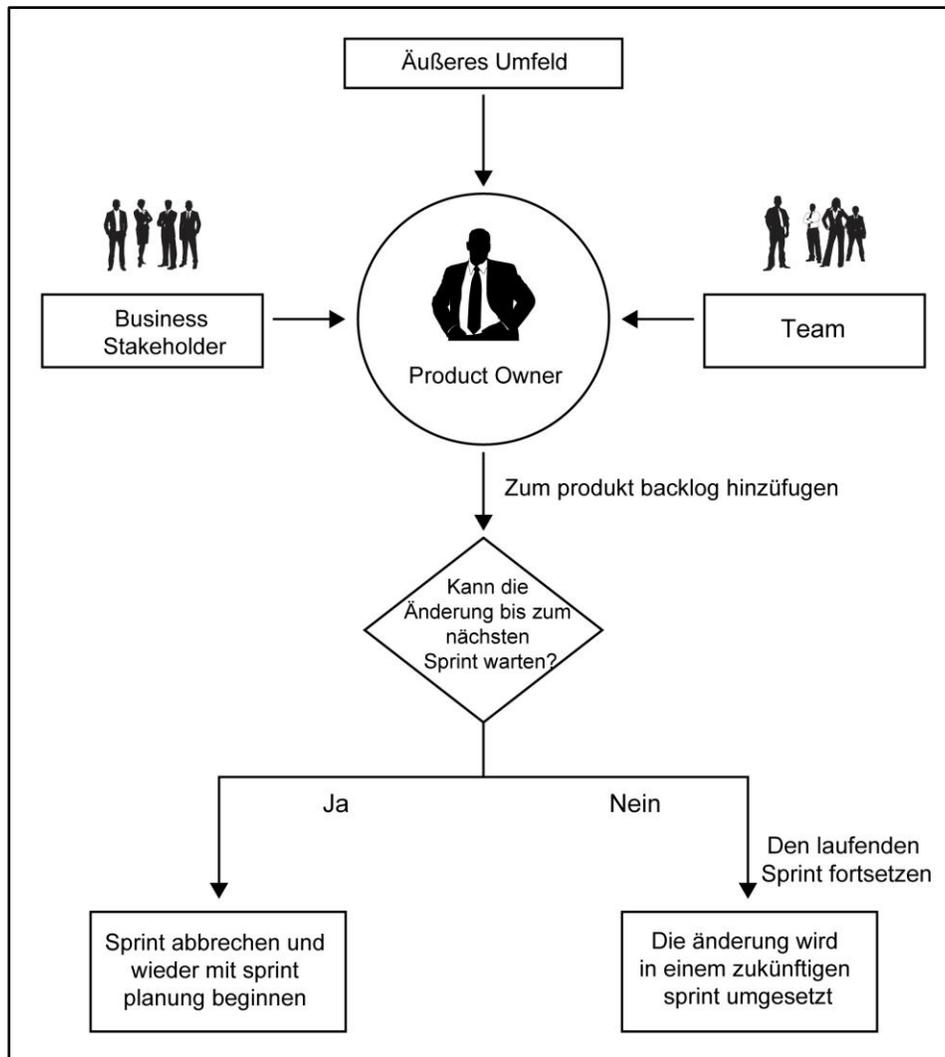


Abbildung 6-6: Integration von Änderung in Scrum

Es gibt nur eine Ausnahme dieser Regel, den Umfang eines Sprint nach Beginn nicht mehr zu ändern. Wenn das Scrum Team feststellt, dass der Arbeitsaufwand deutlich überschätzt wurde und die Kapazität zur Aufnahme weiterer User Storys gegeben ist, kann das Team den Product Owner fragen, welche zusätzlichen User Storys in den laufenden Sprint eingefügt werden sollen.

Durch die Festsetzung des Umfangs eines jeden Sprints ist es dem Team möglich, seine Arbeit und seinen Arbeitsaufwand effizient zu optimieren und zu verwalten. Ein weiterer Vorteil ist, dass das Team sich nicht über das Änderungsmanagement sorgen muss, nachdem es die Arbeit an einem Sprint angefangen hat. Dies ist ein großer Vorteil des Scrum Rahmens im Vergleich zum traditionellem Projekt Management.

Im traditionellen Projekt Management können Änderungen jederzeit innerhalb des Projekt- Entwicklungszyklus angefragt und akzeptiert werden. Oftmals führt dies zu Verwirrung für die Projekt Team-Mitglieder, vermindert die Team Motivation infolge der Diskontinuität und endet in einem Mangel an Konzentration; das Team hat das Gefühl, dass „nichts jemals fertig wird“. Andererseits sind in Scrum Projekten keine Änderungen erlaubt, sobald ein Sprint gestartet wurde. Dies stellt sicher, dass in jedem Sprint zu liefernde Ergebnisse und Aufgaben erfüllt werden. Außerdem erkennt das Unternehmen konkrete Vorteile der lieferfähigen abzuliefernden Ergebnissen am Ende jedes Sprints.

Zudem definieren der Product Owner und die Business Stakeholdern Änderungen und Anforderungen während der dafür vorgesehenen Prozesse *Epic(s) entwickeln*, *Priorisiertes Produkt Backlog erstellen* und *Priorisiertes Produkt Backlog verfeinern*, da sie sich bewusst sind, dass Änderungen während eines Sprints nicht erlaubt sind und ein Sprint nicht mehr als 1 und 4 Wochen dauert.

6.5.1.1 Der Einfluss der Wahrscheinlichkeit von Änderungen auf die Sprint-Länge

Da Änderungen während eines Sprints nicht erlaubt sind, kann die Auswirkung und Häufigkeit der zu erwarteten Änderungen einen Einfluss auf die Entscheidung bzgl. der Länge des Sprints haben, die während des *Release Planung durchführen* Prozesses festgesetzt wird.

Abbildung 6-7 stellt die Auswirkung der Wahrscheinlichkeit von Änderungen auf die Sprint-Länge dar.

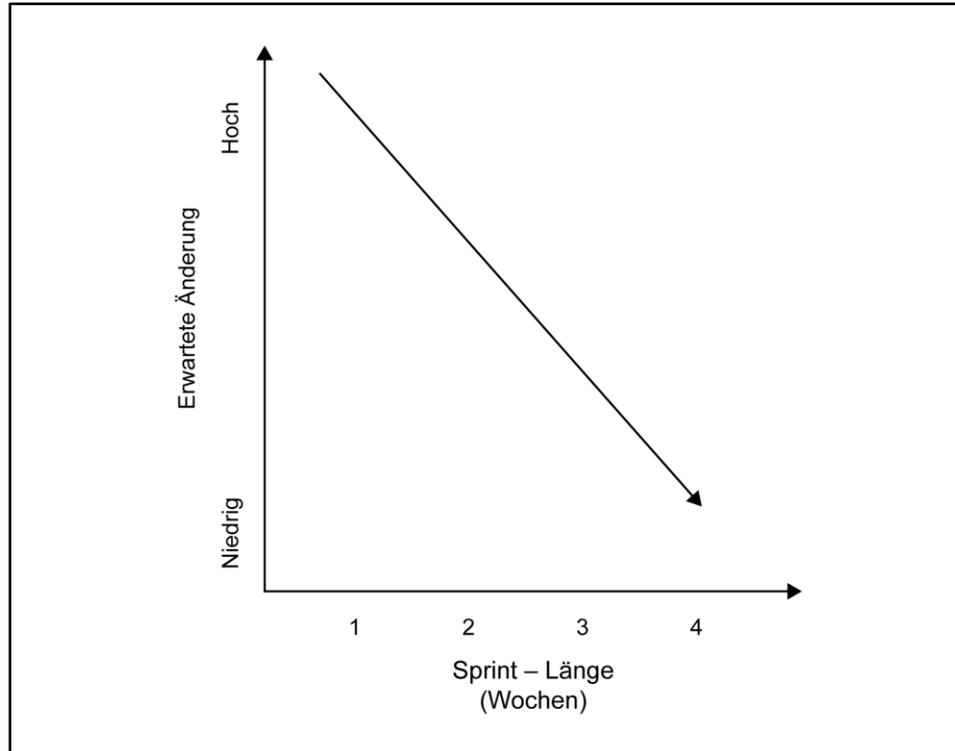


Abbildung 6-7: Auswirkung der Wahrscheinlichkeit von Änderungen auf die Sprint-Länge

Wenn Produktanforderungen nicht sehr klar definiert wurden oder wenn wichtige Änderungen in der nahen Zukunft erwartet werden, wird die Sprint-Länge normalerweise auf ein bis drei Wochen festgelegt. Dies bietet dem Scrum Team die nötige Stabilität an kurzen Sprints zu arbeiten und Ergebnisse zu liefern, welche vom Product Owner und den Business Stakeholdern am Ende des Sprints ausgewertet werden können. Außerdem bietet es genug Flexibilität, um die Anforderungen zu klären und am Ende jedes Sprints Änderungen am Priorisierten Produkt Backlog vorzunehmen.

Wenn allerdings Projektanforderungen generell stabil sind und größere Änderungen in der näheren Zukunft nicht erwartet werden, kann die Länge eines Sprints auf bis zu 6 Wochen festgesetzt werden. Dies bringt den Scrum Team-Mitgliedern die Stabilität, für längere Zeitperioden an den Anforderungen aus dem Priorisierten Produkt Backlog zu arbeiten, ohne die Notwendigkeit, die Prozesse *User Storys Erstellen*; *User Storys abschätzen und User Storys zusagen*, *Aufgaben Identifizieren*, *Aufgaben abschätzen* und andere verwandte Prozesse, die in jedem Sprint durchgeführt werden, mehrfach durchzuführen.

Jedoch ist es wichtig zu vermerken, dass eine erwartete Veränderung nur ein Faktor ist, der benutzt wird, um die Länge eines Sprints festzulegen. Andere Faktoren, die beachtet werden müssen, beinhalten:

- Die tatsächliche Zeitspanne zur Fertigstellung der Arbeit (wenn das Projekt oder das gemeinsame Umfeld eine bestimmte (minimale) Zeit benötigt, um die Aufgabe fertigzustellen, kann das die Sprint-Länge bestimmen)
- Das geplante Release Datum (die Sprint-Länge sollte das Release Datum für das gesamte Produkt oder die Dienstleistung berücksichtigen)
- Jeder andere vom Product Owner oder Scrum Master ermittelte Faktor muss bei der Festlegung der Sprint-Länge berücksichtigt werden.

Es ist wichtig festzuhalten, dass die Sprint-Länge nicht leichtfertig oder regelmässig geändert werden sollte (z.B. ist es nicht empfehlenswert, eine Sprint-Länge von 3 Wochen in einem Sprint zu haben, 2 Wochen im nächsten, 4 Wochen für den dritten usw.). Die Sprint-Länge sollte vorzugsweise konstant sein. Eine der grössten Auswirkungen einer Änderung der Sprint-Länge ist, dass sie ein Rücksetzen aller Werte der Nachverfolgung des Projektfortschritts nach sich zieht. Vorherige Aussagen zur Geschwindigkeit des Teams können gfls. nicht mehr zur Vorhersage und Planung zukünftiger Sprints verwendet werden. Ohne ein akkurate Geschwindigkeit (eine wichtige Metrik in jedem Scrum Projekt) kann das hinsichtlich seiner Effektivität gemessen werden. Ebenso tut sich das Team dann auch schwer, während der Planung des nächsten Sprints die richtige Anzahl von User Storys zusagen. D.h., wenn die Sprint-Länge einmal festgelegt ist, sollte sie möglichst während des gesamten Projekts oder wenigstens über mehrere Sprints hinweg konstant gehalten werden.

6.5.1.2 Änderungen durch Pflegen des Priorisierten Produkt Backlog managen

Ein typisches Priorisiertes Produkt Backlog enthält alle User Storys, deren Zeitabschätzungen (alle überarbeiteten Schätzungen eingeschlossen) und den Status der Anforderungen höherer Priorität. Jede neue oder überarbeitete User Story als Folge von Änderungen der Geschäftlichen Anforderungen, Kundenanfragen, der externen Marktlage, und/oder erlernter Kenntnisse aus früheren Sprints werden auch eingearbeitet.

Eine Hauptverantwortung des Product Owner ist die Pflege des Priorisierten Produkt Backlog. Um sicherzustellen, dass priorisierte Anforderungen im Priorisierten Produkt Backlog, die in die nächsten zwei bis drei Sprints einfließen, in passende User Storys verfeinert werden, wird empfohlen, dass der Product Owner in jedem Sprint genügend Zeit in die Pflege des Priorisierten Produkt Backlog investiert. Der Product Owner ist für das Hinzufügen und Überarbeiten von Priorisierten Produkt Backlog Einträgen verantwortlich und auch dafür, detailliertere User Storys bereitzustellen, die im nächsten Sprint benutzt werden.

Diese Pflege hilft zu gewährleisten, dass die Verfeinerung der Anforderungen und deren User Storys rechtzeitig vor dem Sprint Planungs-Meeting geschieht, so dass das Team eine Menge gut analysierter und klar definierter Storys hat, die einfach in Aufgaben geteilt und anschließend abgeschätzt werden können. Basierend auf erlernten Kenntnissen aus dem laufenden Sprint können Änderungen der Anforderungen oder eine neue Priorisierung entstehen, die leicht in nachfolgende Sprints eingebaut werden können. Diese Pflege unterstützt und erhöht die Flexibilität des Scrum Modells durch Hinzufügen der neusten geschäftlichen und technischen Erkenntnisse in zukünftige Sprints.

Ein Produkt Backlog Review Meeting (auch als Priorisiertes Produkt Backlog Pflege-Session bezeichnet) ist ein formelles Meeting während des „Priorisiertes Produkt Backlog verfeinern“ Prozesses, das dem Scrum Team hilft, das Priorisierte Produkt Backlog zu überprüfen und darüber Konsens zu erreichen.

Außer dem Priorisiertes Produkt Backlog Review Meeting sollte die Pflege des Priorisierten Produkt Backlogs das ganze Projekt hindurch erfolgen und auch Situationen einschließen, in denen der Product Owner neue User Storys schreibt oder das bestehende Priorisierte Produkt Backlog neu priorisiert, Scrum Team-Mitglieder oder Business Stakeholdern dem Product Owner neue User Storys vorschlagen und so weiter.

Es ist wichtig anzumerken, dass jeder Eintrag im Priorisierten Produkt Backlog jederzeit zur Neubeurteilung frei ist, bis der Sprint Backlog im *Sprint Backlog aktualisieren* Prozess festgelegt wurde. Danach können weiterhin Änderungen vorgenommen werden, wenn nötig bis unmittelbar vor dem Sprint Planungsmeeting.

6.5.1.2.1 Effektives Produkt Backlog Review Meeting (oder Priorisiertes Produkt Backlog Pflege-Session)

Der Product Owner übernimmt die Führung im Produkt Backlog Review Meeting, welches während des *Priorisiertes Produkt Backlog verfeinern* Prozesses durchgeführt wird. Es ist wichtig, dass der Product Owner die Ziele setzt und idealerweise eine Agenda entwickelt, bevor das Backlog Review Meeting beginnt. Andernfalls wird die Sitzung unstrukturiert und mag sich als unproduktiv erweisen. Außerdem ist es wichtig, die Anzahl der an dem Meeting teilnehmenden Business Stakeholdern zu begrenzen. Zu viele Teilnehmer zu haben birgt die Gefahr, die Effizienz des gesamten Meetings zu mindern. Der Product Owner sollte nur die Business Stakeholdern einladen, deren Feedback für die Pflege-Session benötigt wird. Alle Scrum Team-Mitglieder sollten einbezogen werden, da ihr Input wertvoll für die zu leistende Arbeit und jedes auftretende Problem ist. Wenn die Pflege-Session in einer neuen Priorisierung oder einer Änderung im Priorisierten Produkt Backlog resultiert, ist es wichtig, dass das Team mit den Änderungen einverstanden ist.

Eine erfolgreiche Pflege-Session sollte zu klar definierten Priorisierten Produkt Backlog Einträgen (PBIs) führen, so dass das Scrum Team die Kundenanforderungen klar versteht. Dies hilft dem Team, sich mit allen User Storys vertraut zu machen, falls eine oder mehrere davon kurzfristig in einen Sprint eingebaut werden müssen. Auch Akzeptanz- und Done-Kriterien können während der Pflege-Session besprochen werden. Pflēgetätigkeiten sind nicht time-boxed. Die Pflege des Priorisierten Produkt Backlogs ist eine kontinuierliche Aufgabe für den Product Owner.

6.5.1.3 Veränderungsmanagement während der Sprint Vorführung und Validierung

Auch wenn die Business Stakeholdern und der Product Owner das letzte Wort bezüglich der Einträge des Priorisierten Produkt Backlogs und auch hinsichtlich der Akzeptanz oder Ablehnung von Change Requests während des *Sprint Vorführen und Validieren* Prozesses haben, ist es die Verantwortung des Scrum Masters sicherzustellen, dass die Anforderungen und Abnahmekriterien für die im laufenden Sprint fertiggestellten User Storys während des Sprint Review Meetings nicht geändert wurden. Dies verhindert, dass fertiggestellte User Storys nicht aufgrund von nicht erfüllten neuen Anforderungen abgelehnt werden. Wenn irgendwelche Anforderungen geändert werden müssen, erfordert dies, dazugehörige PBIs zu überarbeiten, um die modifizierten Anforderungen in folgenden Sprints aufzunehmen.

6.6 Änderungen in Programmen und Portfolios

Jegliche Änderungen, die in den Programmen oder Portfolios aufkommen, können einen Dominoeffekt auch auf alle abhängige Projekte und Sprints haben. Deshalb ist es ratsam, auf diesem hohen Level Änderungen auf ein Minimum zu beschränken. Wenn eine Änderung notwendig ist und alle Business Stakeholdern übereinstimmen, die Veränderung auf diesem Level durchzuführen, sollte folgendes berücksichtigt werden.

6.6.1 Im Portfolio

1. Es ist nicht ratsam, Änderungen zwischen zwei Portfolio Backlog Meetings vorzunehmen.
2. Wenn die Änderung minimal ist, sollte der Portfolio Product Owner die Zustimmung aller relevanten Business Stakeholdern (z.B. Sponsor, Kunde und Endanwender) sicherstellen und dann die Anforderungen dem Portfolio Backlog hinzufügen. Die Product Owner des Programms und Projekts werden in Betracht ziehen, diese Anforderungen in zukünftige Sprints aufzunehmen.
3. Wenn es um eine größere Änderung geht, müssen die Arbeiten am Portfolio sowie den dazugehörigen Programmen, Projekten und Sprints gestoppt und die nächsten Schritte in einem Portfolio Backlog Meeting besprochen werden.
4. Portfolio Priorisiertes Produkt Backlog Meetings (auch Portfolio Backlog Meetings genannt), sollten in 4- bis 12-Monatsintervallen abgehalten werden. Die Häufigkeit und Auswirkung von Änderungen auf das Portfolio bestimmt größtenteils die Zeitspanne zwischen zwei Portfolio Backlog Meetings. Wenn mehrere Änderungen am Portfolio erwartet werden, ist es vorzuziehen, Portfolio Backlog Meetings regelmäßiger abzuhalten (z.B. alle 4–6 Monate); wenn weniger Änderungen erwartet werden und die Anforderungen stabil bleiben, kann der Abstand zwischen zwei Portfolio Backlog Meetings vergrößert werden (z.B. alle 9–12 Monate).

6.6.2 Im Programm

1. Es wird nicht empfohlen, Änderungen zwischen zwei Programm Backlog Meetings vorzunehmen.
2. Wenn die Änderung minimal ist, sollte der Programm Product Owner die Zustimmung aller relevanten Business Stakeholdern (z.B. Sponsor, Kunde und Endanwender) und dem Portfolio Product Owner sicherstellen und dann die Anforderungen dem Programm Backlog hinzufügen. Die Product Owner des Projekts werden in Betracht ziehen, diese Anforderungen in zukünftige Sprints aufzunehmen.
3. Wenn es um eine größere Änderung geht, müssen die Arbeiten am Programm sowie den dazugehörigen Projekten und Sprints gestoppt und die nächsten Schritte in einem Priorisierten Produkt Backlog Meeting besprochen werden.
4. Programm Priorisiertes Produkt Backlog Meetings (auch Programm Backlog Meetings genannt), sollten in 2- bis 6-Monatsintervallen abgehalten werden. Die Häufigkeit und Auswirkung von Änderungen auf das Programm bestimmt größtenteils die Zeitspanne zwischen zwei Programm Backlog Meetings.

Wenn mehrere Änderungen am Programm erwartet werden, ist es vorzuziehen, Programm Backlog Meetings regelmäßiger abzuhalten (z.B. alle 2–3 Monate); wenn weniger Änderungen erwartet werden und die Anforderungen stabil bleiben, kann der Abstand zwischen zwei Programm Backlog Meetings vergrößert werden (z.B. alle 5–6 Monate). Abbildung 6-8 veranschaulicht, wie Änderungen im Scrumfluss für Portfolios und auch für Programme gehandhabt werden können.

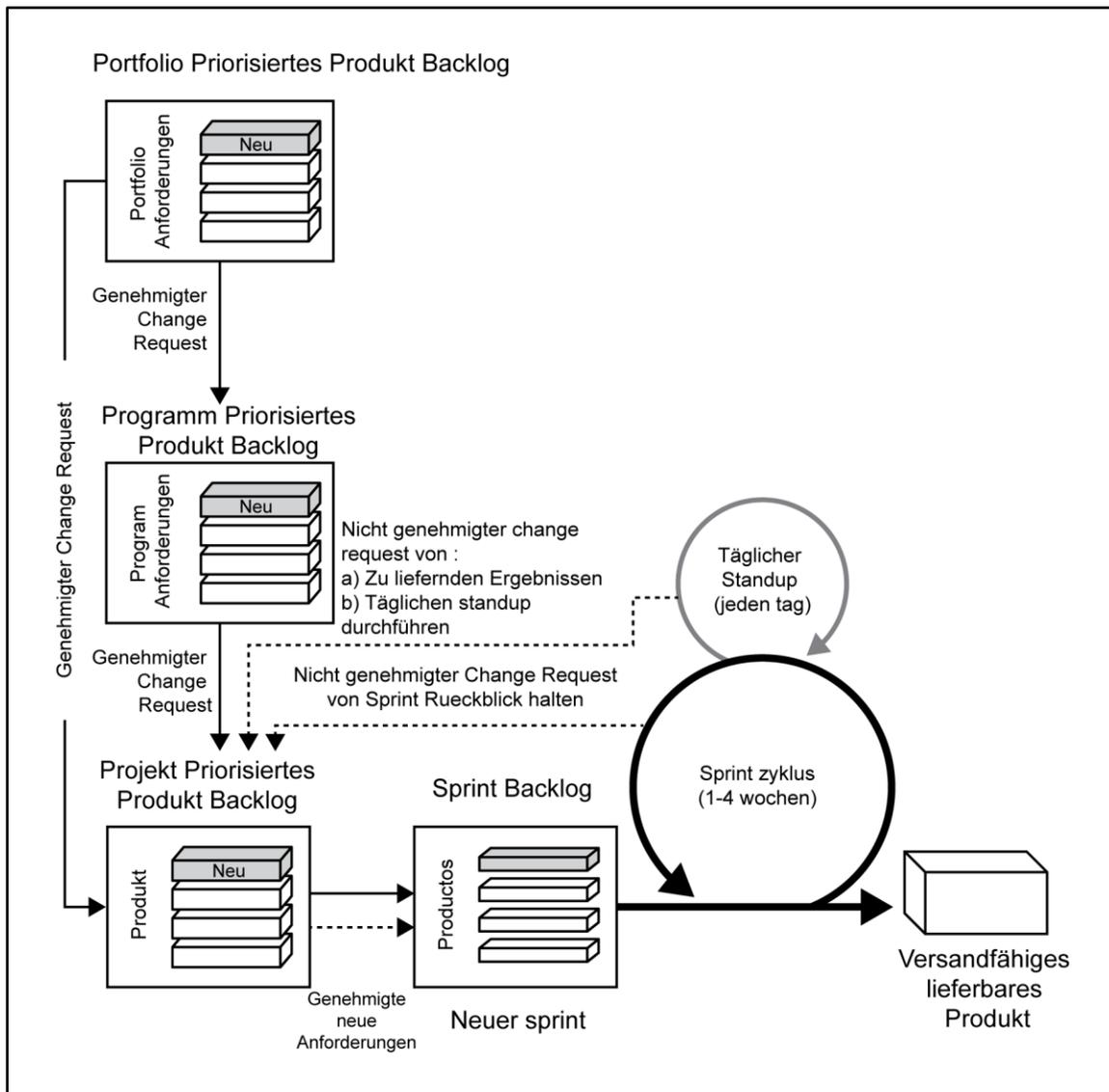


Abbildung 6-8: Änderungen im Portfolio und Programm einbinden

6.7 Zusammenfassung der Verantwortlichkeiten

Kernrolle	Verantwortlichkeiten
Scrum Guidance Body	<ul style="list-style-type: none"> Bietet während des ganzen Projekts allgemeine Anleitung für die Änderungsmanagement Prozeduren, die befolgt werden müssen
Portfolio Product Owner	<ul style="list-style-type: none"> Liefert Change Requests auf dem Portfolio Level Genehmigt Produkte die den Portfolio Anforderungen entsprechend berichtigt, entfernt oder hinzugefügt wurden
Portfolio Scrum Master	<ul style="list-style-type: none"> Fördert Identifikation, Bewertung und Verwaltung der Change Requests auf dem Portfolio Level
Programm Product Owner	<ul style="list-style-type: none"> Liefert Change Requests auf dem Programm Level Genehmigt Produkte, die den Programm Anforderungen entsprechend berichtigt, entfernt oder hinzugefügt wurden
Programm Scrum Master	<ul style="list-style-type: none"> Fördert Identifikation, Bewertung und Verwaltung der Change Requests auf dem Programm Level
Business Stakeholdern	<ul style="list-style-type: none"> Liefert Anfragen auf Änderungen Ist involviert in Genehmigen und Priorisieren von Change Requests
Product Owner/Chief Product Owner	<ul style="list-style-type: none"> Liefert Anfragen auf Änderungen in einem Projekt Bewertet die Auswirkungen der Change Requests auf das Portfolio, Programm oder Projekt Priorisiert User Storys im Projekt Priorisierten Produkt Backlog Bewertet die Auswirkungen von vom Scrum Team identifizierten Problemen auf das Projektziel Versorgt die Business Stakeholdern mit klaren Informationen repriorisierten Einträgen im Priorisierten Produkt Backlog Artikeln
Scrum Master/Chief Scrum Master	<ul style="list-style-type: none"> Fördert die Identifikation, Bewertung und Eskalation von Problemen und Change Requests durch das Scrum Team
Scrum Team	<ul style="list-style-type: none"> Schlägt Verbesserungen oder Änderungen während der Prozesse <i>Zu liefernde Ergebnisse erzeugen und Täglichen Standup durchführend</i>

Tabelle 6-1: Zusammenfassung der Verantwortlichkeiten hinsichtlich Änderungen

6.8 Scrum versus traditionelles Projekt Management

Das Änderungsmanagement in traditionell gemanagten Projekten ist eng mit dem Konfigurationsmanagement verbunden. Alle Änderungen werden nach ihrem Ausmaß an Variationen anhand eines Basiswerts ermittelt. Dem Projekt Manager wird ein Spielraum gegeben, innerhalb dessen er/sie alltägliche Vorgänge und Entscheidungen des Projekts managen kann. Wenn ein Change Request den vorgegebenen Spielraum übersteigt, muss der Projekt Manager die vorgeschlagene Änderung an einen höheren Managementlevel weiterreichen und deren Entscheidung abwarten, bevor die Änderung eingearbeitet werden kann. Der Projekt Manager loggt die Änderungsanfrage in ein Problem Log oder Änderungslog und gibt dann die Änderung an eine höhere Instanz weiter. Dies kann der Sponsor des Projekts sein, oder irgendein anderer relevanter Business Stakeholdern und Entscheidungsträger. Zu irgendeinem Zeitpunkt wird eine Beurteilung der Auswirkung durchgeführt und basierend auf den geschätzten Auswirkungen der Änderung wird entschieden, ob die Änderung eingearbeitet wird oder nicht. Auch der Projekt Manager kann zu jedem Problem, das die Änderung darstellt, mögliche Lösungen vorschlagen. Wenn die höhere Instanz die Entscheidung gefällt hat, die Änderung durchzuführen, ist der Projekt Manager dafür verantwortlich, die Änderung ordnungsgemäß einzubauen.

Verglichen mit dem traditionellen Projekt Management verlaufen Änderungen in Scrum ganz anders. Der Scrum Rahmen ist hochgradig darauf eingestellt, Änderungen effektiv und effizient zu handhaben. Wann immer der Product Owner oder das Scrum Team ein Problem bzw. einen Fehler erkennen oder ein Priorisierter Produkt Backlog Eintrag ergänzt, ersetzt oder hinzugefügt werden muss, wird die Änderung ins Priorisierte Produkt Backlog eingebracht. Ebenso kann das Senior Management, der Product Owner oder Business Stakeholdern Change Requests zu dem Priorisierten Produkt Backlog hinzufügen. Product Owner und Business Stakeholdern genehmigen Change Requests und priorisieren das Backlog dementsprechend neu. Wann immer ein Problem oder eine neue Anforderung unmittelbar bearbeitet werden muss und eine zwingende Änderung zur Folge hat, die einen laufenden Sprint beeinflusst, beendet der Product Owner mit der Zustimmung relevanter Business Stakeholdern den Sprint. Sobald der Sprint beendet wurde, wird er neu geplant und mit den neu eingebauten Anforderungen erneut gestartet.

Falls das Problem oder die Änderung nicht bedeutend ist und keine Änderung am laufenden Sprint nach sich zieht, wird die Änderung dem Priorisierten Produkt Backlog hinzugefügt und in die Planung eines nachfolgenden Sprints eingearbeitet. Dies gibt Business Stakeholdern die Möglichkeit, auf Änderungen im externen Umfeld zu reagieren, während sie ein gewisses Maß an Kontrolle über die laufenden Tätigkeiten innerhalb des Projekts beibehalten. Am Ende jedes Sprints werden fertiggestellte zu liefernde Ergebnisse vom Scrum Team vorgeführt. Diese potenziell versandfähigen zu liefernden Ergebnisse können vom Product Owner und anderen Business Stakeholdern überprüft werden.

7. RISIKO

7.1 Einführung

Das Ziel dieses Kapitels ist es, das Konzept des Risikos zu definieren, die Handhabung von Risiken bei einem Scrum-Projekt zu besprechen und die Hilfsmittel, welche den Umgang mit Risiken erleichtern, aufzuzeigen. Um die Lebensfähigkeit eines Unternehmens zu sichern, die Wahrscheinlichkeit eines Scheiterns des Projekts zu verringern, und fundiertere geschäftliche Entscheidungen zu treffen, ist es wichtig dass Risiken effektiv und in einer gut organisierten und methodischen Art und Weise gehandhabt werden.

Bei einem Scrum-Projekt werden Risiken prinzipiell minimiert, zu einem grossen Teil dadurch, dass die Arbeit in Sprints aufgeteilt ist, wodurch eine kontinuierliche Serie von zu liefernden Ergebnissen in kurzen Zyklen produziert wird, diese mit den Erwartungen abgeglichen werden und der Product Owner aktiv am Projekt beteiligt ist. Aber selbst im einfachsten aller Projekte können Dinge schief gehen, weshalb es wichtig ist, eine Strategie zur Identifizierung und Behebung von Risiken zu haben.

Risiko, nach der Definition des *Scrum Body of Knowledge (SBOK®)*, kann auf folgende Sachverhalte angewendet werden:

- Portfolios, Programme, und/oder Projekte in *jeder* Branche.
- Produkte, Dienstleistungen, oder andere Ergebnisse, welche an die Business Stakeholdern geliefert werden.
- Projekte jeglicher Größe und Komplexität

Der Begriff "Produkt" bezieht sich im Rahmen dieses Body of Knowledges auf ein Produkt, eine Dienstleistung, oder ein anderes zu lieferndes Ergebnis. Scrum kann effektiv auf jedes Projekt in jeder Branche angewendet werden - von kleinen Projekten oder Teams mit eventuell nur sechs Teammitgliedern hin zu großen, komplexen Projekten mit bis zu mehreren hundert Teammitgliedern.

Das Kapitel unterteilt sich in folgende Abschnitte:

7.2 Leitfaden für Kernrollen—Dieser Abschnitt bietet eine Anleitung dazu, welche Bereiche relevant für jede Scrum Kern-Rolle sind: Product Owner, Scrum Master, and Scrum Team.

7.3 Was ist Risiko?—Dieser Abschnitt definiert das Konzept des Risikos und erklärt, wie es die Ziele eines Projekts beeinflussen und zum Erfolg oder Scheitern eines Projekts beitragen kann.

7.4 Verfahrensweise beim Risiko Management—Dieser Abschnitt offeriert Schlüsseltechniken für den Umgang mit Risiken und arbeitet heraus, wie sich Strategien entwickeln lassen, um Risiken zu identifizieren, zu bewerten und zu handhaben.

7.5 Mit Scrum Risiken verringern—Dieser Abschnitt erklärt die Schlüsselaspekte von Scrum, welche es zu einer idealen Management -Methode für das effektive Handhaben von Risiken auf unterschiedlichsten Ebenen machen— Portfolio, Programm, und Projekt.

7.6 Übersicht der Verantwortlichkeiten—Dieser Abschnitt beschreibt die Verantwortlichkeiten jeder Person oder Rolle innerhalb eines Projekts im Bezug auf Risiko Management.

7.7 Scrum versus traditionelles Projektmanagement—Dieser Abschnitt behandelt die Vorteile der Handhabung von Risiken mit Hilfe von Scrum Methoden gegenüber der Methoden der traditionellen Projektmanagement-Modelle.

7.2 Leitfaden für Kernrollen

1. Product Owner—Die Hauptverantwortung für das Handhaben von Risiken bei einem Projekt trägt der Product Owner; aus diesem Grund betrifft das gesamte Kapitel überwiegend diese Rolle.
2. Scrum Master—Der Scrum Master sollte sich mit dem gesamten Kapitel vertraut machen, besonders jedoch mit den Abschnitten 7.3, 7.4, und 7.7.
3. Scrum Team—Das Scrum Team sollte sich überwiegend mit den Abschnitten 7.3 und 7.7 beschäftigen.

7.3 Was ist Risiko?

Risiko wird als ungewisses Ereignis definiert, das die Ziele eines Projekts beeinflussen und zum Erfolg oder Scheitern des Projekts beitragen kann. Risiken, welche das Potential besitzen, das Projekt positiv zu beeinflussen, werden Gelegenheiten genannt, während Gefahren die Risiken darstellen, welche ein Projekt negativ beeinflussen können. Risiken müssen vorausschauend gehandhabt werden, ein iterativer Prozess, der beim Anfang des Projekts beginnen und sich durch das gesamte Projekt hindurch fortsetzen sollte. Der Prozess des Risikomanagements sollte einigen standardisierten Schritten folgen, um sicherzustellen, dass Risiken als solche erkannt und bewertet werden und die korrekte Vorgehensweise formuliert und in entsprechender Form implementiert werden kann.

Risiken sollten vorrangig auf Basis von zwei Faktoren identifiziert, bewertet und bearbeitet werden: die Wahrscheinlichkeit des Eintretens und die wahrscheinlichen Auswirkungen im Falle des Eintretens. Risiken mit hoher Wahrscheinlichkeit und hohen Auswirkungen sollten vor solchen mit geringeren Bewertungen behandelt werden. Im Allgemeinen ist es nach dem Feststellen eines Risikos wichtig, die grundsätzlichen Aspekte des Risikos, seine möglichen Ursachen, den Bereich der Unsicherheit, und die möglichen Auswirkungen im Falle des Eintretens des Risikos zu verstehen.

7.3.1 Der Unterschied zwischen Risiken und Problemen

Risiken sind die Ungewissheiten im Rahmen eines Projekts, welche das Ergebnis des Projekts sowohl zum Positiven als auch zum Negativen maßgeblich beeinflussen können. Da Risiken Ungewissheiten in der Zukunft darstellen, haben sie keine gegenwärtigen Auswirkungen auf das Projekt, können diese aber potentiell in der Zukunft haben. Im Folgenden sind einige Beispiele für Risiken wiedergegeben:

- Selbst nach ausgiebiger Schulung kann es sein, dass die Kundendienstmitarbeiter noch nicht in der Lage sind, am Stichtag Aufträge zu bearbeiten.
- Die Malerarbeiten können durch starke Regenfälle verzögert werden, was sich wiederum negativ auf den Zeitplan des Projekts auswirken kann.

Probleme sind prinzipiell klar definierte Gewissheiten, welche das Projekt gegenwärtig beeinflussen: daher besteht, anders als beim Risiko, keine Notwendigkeit eine Einschätzung der Wahrscheinlichkeit vorzunehmen. Probleme müssen erledigt werden. Im Folgenden sind einige Beispiele für Probleme wiedergegeben:

- Die Finanzierung wird nicht bewilligt.
- Anforderungen sind nicht klar.

Risiken, welche nicht rechtzeitig angegangen werden, können zu Problemen werden. Das Ziel von Risikomanagement ist es, so vorbereitet zu sein, dass Pläne vorhanden sind, um mit allen möglicherweise auftretenden Risiken fertig zu werden.

7.3.2 Risikoverhalten

Business Stakeholdern sind all diejenigen Personen oder Organisationen, welche durch ein Projekt berührt werden, aber auch diejenigen, welche die Möglichkeit haben das Projekt zu beeinflussen. Es ist wichtig, das Risikoverhalten von Business Stakeholdern zu verstehen. Risikoverhalten wird von diesen drei Faktoren beeinflusst:

1. Risikobereitschaft: bezieht sich darauf, wie viel Ungewissheit der Business Stakeholdern oder die Organisation bereit ist zu tragen.
2. Risikotoleranz: bezieht sich auf den Grad, die Menge und das Ausmaß von Risiken, welche die Business Stakeholdern aushalten können.
3. Risikoschwellwert: bezieht sich auf das Level des Risikos welches für die Gemeinschaft der Business Stakeholdern akzeptabel ist. Ein Risiko ist entweder über oder unter dem Risikoschwellwert einzuordnen. Wenn es darunter liegt, sind die Business Stakeholdern oder die Organisation eher dazu bereit, das Risiko zu akzeptieren.

Grundsätzlich bestimmt das Risikoverhalten von Business Stakeholdern, wie viel Risiko sie als akzeptabel einschätzen, und damit ab welchen Punkt sie zu Maßnahmen greifen, um potentielle negative Auswirkungen des Risikos zu vermindern. Deshalb ist es wichtig, die Toleranzgrenzen der Business Stakeholdern in Verbindung mit einer Vielzahl von Faktoren, wie beispielsweise Qualität, Umfang und Zeitplan, zu verstehen.

Die *Nutzenfunktion* ist ein Modell, welches man nutzt um die Risikoneigung von Business Stakeholdern oder ihre Haltung gegenüber von Risiken zu messen. Es definiert die Grenzen und Bereitschaft von Business Stakeholdern, Risiken zu akzeptieren. Die Nutzenfunktion hat drei Kategorien:

1. Risikoscheu: Der Business Stakeholdern ist nicht bereit, ein Risiko zu akzeptieren, egal welcher Nutzen oder welche Gelegenheit dabei erwartet wird.
2. Risikoneutral: der Business Stakeholdern ist weder risikoscheu noch risikosuchend, und jedwede Entscheidung wird nicht durch den Grad an Ungewissheit des Ergebnisses beeinflusst. Wenn zwei mögliche Szenarios gleichviel Gewinn mit sich bringen, wird der risikoneutrale Business Stakeholdern nicht besorgt sein, wenn eines der Szenarien riskanter als das andere ist.
3. Risikosuchend: Der Business Stakeholdern ist bereit, Risiken zu akzeptieren, selbst wenn sie nur einen geringfügigen Anstieg von Ertrag oder Gewinn für das Projekt mit sich bringen.

7.4 Verfahrensweise beim Risikomanagement

Risikomanagement besteht aus den folgenden fünf Schritten, die iterativ während des gesamten Projekts durchgeführt werden:

1. Risikoidentifikation: Die Verwendung von verschiedenen Methoden zur Identifizierung eines potentiellen Risikos
2. Risikobewertung: Bewertung und Einschätzung des erkannten Risikos
3. Risikopriorisierung – Priorisierung von Risiken, welche für eine spezifische Maßnahme in das Priorisierte Produkt Backlog aufgenommen werden.
4. Risikominderung: Entwicklung einer angemessenen Strategie, um mit dem Risiko fertig zu werden
5. Risikokommunikation: Vermittlung der Ergebnisse der ersten vier Schritte an die entsprechenden Business Stakeholdern und Ermittlung ihrer Sichtweise im Bezug auf ungewisse Ereignisse.

7.4.1 Risikoidentifikation

Die Mitglieder des Scrum Teams sollten versuchen, alle Risiken, die mögliche Auswirkungen auf das Projekt haben können, zu identifizieren. Dies kann nur dann gründlich erfolgen, wenn das Projekt aus verschiedenen Blickwinkeln und mit einer Vielzahl von Methoden betrachtet wird. Risikoidentifikation geschieht während der gesamten Dauer des Projekts und identifizierte Risiken werden anhand mehrerer Scrum Prozesse bearbeitet, darunter unter anderem *Priorisiertes Produkt Backlog erstellen*, *Priorisiertes Produkt Backlog verfeinern* und *Sprint durchführen und validieren*.

Die folgenden Methoden werden oft zur Identifikation von Risiken verwendet.

7.4.1.1 Methoden der Risikoidentifikation

1. Gewonnene Erfahrungen durch Sprint-Rückblicke oder Projekt-Projektrückblicke

Von ähnlichen Projekten und vorhergehenden Sprints im selben Projekt zu lernen und die Ungewissheiten, welche diese Projekte und Sprints beeinflusst haben, zu untersuchen, kann ein guter Weg sein, um Risiken zu erkennen.

2. Risiko-Checklisten

Risiko-Checklisten können Kernpunkte enthalten, welche bei der Identifikation von Risiken beachtet werden sollten, übliche Risiken anführen, welche in Scrum-Projekten auftreten, oder gar Kategorien von Risiken enthalten, welche vom Team beachtet werden sollten. Checklisten sind ein nützliches Mittel, um eine möglichst vollständige Risikoidentifikation sicherzustellen.

3. Risiko-Stimulierungs-Listen

Risiko-Stimulierungs-Listen werden verwendet, um Ideen über die möglichen Quellen von Risiken anzuregen. Risiko-Stimulierungs-Listen für verschiedenste Branchen und Projektarten sind öffentlich verfügbar.

4. Brainstorming

Sitzungen, in welchen die relevanten Business Stakeholdern und Mitglieder des Scrum Kernteams offen ihre Ideen mittels Diskussionen und Wissensaustausch weitergeben. Üblicherweise werden sie durch einen Moderator geleitet.

5. Risikostrukturplan

Eine der Schlüsselmethoden zur Erkennung von Risiken ist ein Risikostrukturplan. In diesem Gefüge werden Risiken auf Basis ihrer Kategorien und Gemeinsamkeiten gruppiert. Risiken können beispielsweise als finanziell, technisch, oder sicherheitsbezogen eingestuft werden. Dies ermöglicht dem Team, das jeweilige Risiko besser einzuplanen und zu bearbeiten.

6. Interviews

Interviews und informelle Treffen mit Business- oder anderen Business Stakeholdern wie Senior Management, Scrum-Team-Mitgliedern, technischen Experten, Kunden und Endanwendern können helfen, Projektrisiken zu identifizieren.

7.4.2 Risikobewertung

Die Risikobewertung hilft dabei, die möglichen Auswirkungen von Risiken, die Häufigkeit ihres Auftretens, und den Zeitpunkt ihres Auftretens zu verstehen. Die gesamte Wirkung auf den Geschäftswert sollte eingeschätzt werden; wenn dieser die Geschäftliche Rechtfertigung hinreichend deutlich übersteigt, muss eine Entscheidung darüber getroffen werden, ob das Projekt fortgesetzt werden soll.

Die Risikobewertung erfolgt mit Hinblick auf Wahrscheinlichkeit, zeitliche Nähe und Wirkung. Die Risikowahrscheinlichkeit bezieht sich darauf, wie wahrscheinlich es ist, dass ein Risiko eintritt, wohingegen zeitliche Nähe sich auf den Zeitpunkt des möglichen Eintretens des Risikos bezieht. Wirkung bezieht sich auf die wahrscheinlichen Folgen von Risiken für das Projekt oder die Organisation. Um die Wahrscheinlichkeit eines Risikos einzuschätzen, können verschiedenste Techniken verwendet werden, darunter beispielsweise Wahrscheinlichkeitsbäume, die Pareto-Analyse, und ein Wahrscheinlichkeits- / Auswirkungsraster

Neben der Wahrscheinlichkeit kalkuliert Risikobewertung die möglichen Auswirkungen von Risiken auf das Projekt oder die Organisation. Diese können anhand von Methoden wie Risikomodellen und dem Erwarteten Geldwert eingeschätzt werden.

7.4.2.1 Methoden der Risikobewertung

1. Risiko-Treffen

Risiken können einfach durch den Product Owner priorisiert werden, indem dieser ein Treffen des Scrum Kernteams, eventuell erweitert um relevante Business Stakeholdern, einberuft. Das Team kann bei diesem Treffen verschiedene Risiken auf Grund ihrer eigenen subjektiven Bewertung der Auswirkungen dieser Risiken auf die Ziele des Projekts priorisieren.

2. Wahrscheinlichkeitsbäume

Mögliche Ereignisse werden in einem Baumgraphen dargestellt; jedes mögliche Ergebnis eines Ereignisses erhält hierbei einen eigenen Ast. Die Wahrscheinlichkeit jedes Ergebnisses wird am jeweiligen Ast angegeben, diese Werte wiederum können dann zur Kalkulation der gesamten Wirkung im Fall des Eintretens des Risikos bei einem Projekt (vergleiche Abbildung 7-1).

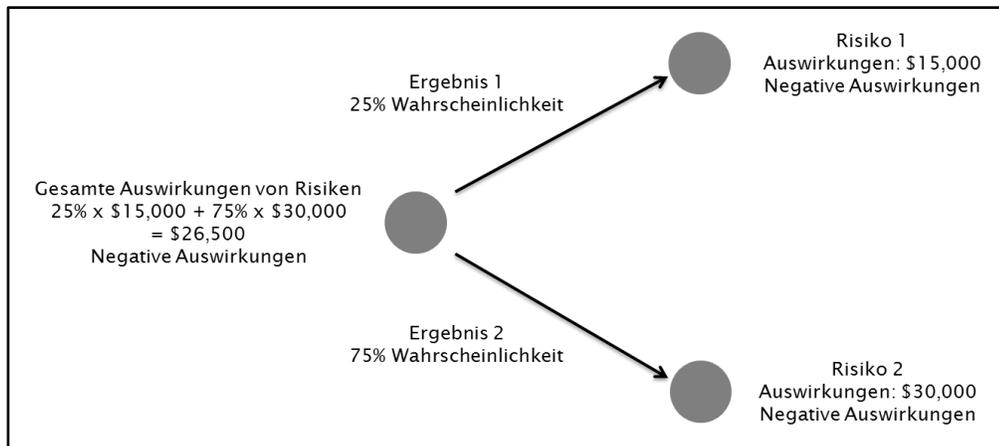


Abbildung 7-1: Beispiel für einen Wahrscheinlichkeitsbaum

3. Pareto-Analyse

Diese Methode der Risikobewertung beinhaltet die Gewichtung von Risiken nach ihrem Ausmaß, und hilft dem Scrum Team dabei, Risiken in der Reihenfolge ihrer möglichen Auswirkung auf das Projekt zu bearbeiten. Zum Vergleich: in der Abbildung 7-2 ist Risiko 1 dasjenige mit den größten Auswirkungen und sollte daher möglichst zuerst behandelt werden.

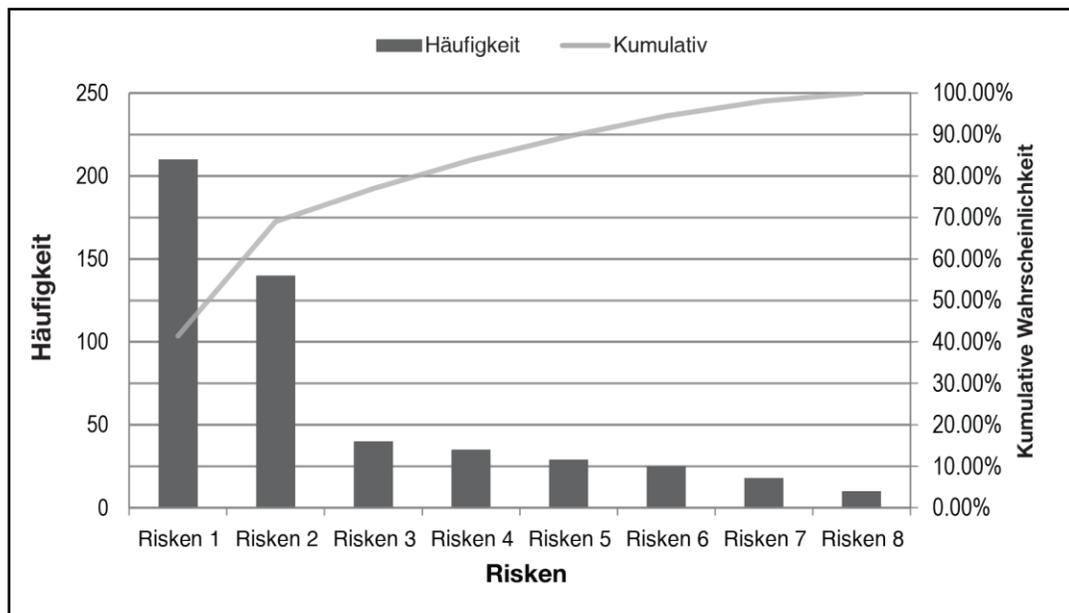


Abbildung 7-2: Beispiel eines Pareto-Diagramms

4. Wahrscheinlichkeits- / Auswirkungsraster

Jedes Risiko wird anhand der Wahrscheinlichkeit des Auftretens und seiner Möglichen Auswirkungen auf die Projektziele bearbeitet. Üblicherweise wird sowohl der Wahrscheinlichkeit als auch den Auswirkungen unabhängig voneinander eine numerische Bewertung zugewiesen. Die beiden Werte werden dann miteinander multipliziert, um die Schwere des entsprechenden Risikos zu bestimmen, welche dann wiederum zur Priorisierung von Risiken genutzt werden kann.

Beispielsweise würde die Kennzahl für die Schwere des Risikos bei einer Risikowahrscheinlichkeit von 50% und einer Auswirkungsbewertung von 0,6 wie folgt kalkuliert:

$$0,5(\text{Wahrscheinlichkeit}) \times 0,6(\text{Auswirkung}) = 0,3$$

Die verwendeten Bewertungsmuster werden innerhalb der Organisation oder für das Projekt bestimmt. Oft werden Dezimalzahlen verwendet, bei denen ein Wert von 0,5 Wahrscheinlichkeit ein Eintreten des Risikos in 50 % aller Fälle indiziert. Andere Varianten wären eine Skala von eins bis zehn, oder die Bewertung Hoch (3), Mittel (2), und Gering (1).

Abbildung 7-3 gibt die Verwendung einer Dezimalskala wieder. Jedes Risiko wird anhand der Wahrscheinlichkeit seines Auftretens und seiner Auswirkungen auf einer objektiven Skala bewertet.

Wahrscheinlichkeits- / Auswirkungsraster						
Wahrscheinlichkeit	Gefahren			Gelegenheiten		
0,90	0,09	0,27	0,72	0,72	0,27	0,09
0,75	0,075	0,225	0,60	0,60	0,225	0,075
0,50	0,05	0,15	0,40	0,40	0,15	0,05
0,30	0,03	0,09	0,24	0,24	0,09	0,03
0,10	0,01	0,03	0,08	0,08	0,03	0,01
	Low 0,1	Medium 0,3	High 0,8	High 0,8	Medium 0,3	Low 0,1

Auswirkungen

 Niedriger PI-Wert	 Moderater PI-Wert	 Hoher PI-Wert
---	---	---

Abbildung 7-3: Beispiel eines Wahrscheinlichkeits- / Auswirkungsrasters

Die Methode der Zuordnung von Werten für Wahrscheinlichkeit und Auswirkungen von Risiken unterscheidet sich je nach Projekt und der Anzahl der bewerteten Risiken, aber auch aufgrund von vorher bestehenden Abläufen und Prozeduren. Dennoch kann mittels dieser einfachen Formel

(Wahrscheinlichkeit x Auswirkungen) der Schwere des Risikos eine sowohl numerische als auch kategoriale Bewertung zugewiesen werden.

5. Erwarteter Geldwert (EMV)

Der monetäre Wert eines Risikos basiert auf dem Erwarteten Geldwert (EMV). Der EMV wird berechnet, indem die Finanziellen Auswirkungen des Risikos mit der vom Kunden eingeschätzten Wahrscheinlichkeit multipliziert wird.

Erwarteter Geldwert = Auswirkung des Risikos (in Dollar) x Risikowahrscheinlichkeit (als Prozentzahl)

Ein Risiko mit einer geschätzten negativen Auswirkung von 1.000 \$ und einer Wahrscheinlichkeit des Eintretens von 50 % resultiert beispielsweise in folgendem EMV:

$$\text{EMV} = 1.000 \$ \times 0,50 = 500 \$$$

7.4.3 Risikopriorisierung

Scrum ermöglicht die schnelle Identifikation und Bewertung von Risiken. Identifizierte Risiken werden bei der Erstellung des Priorisierten Produkt Backlogs oder bei einem Update im Rahmen der Pflege des Priorisierten Produkt Backlogs mit einbezogen—daher kann das Priorisierte Produkt Backlog als ein risikobereinigtes Priorisiertes Produkt Backlog angesehen werden.

Die Risiken können auf Grund von jeglicher zuvor genannten Methode von Risikoidentifikation und Risikobewertung erkannt und eingeschätzt werden.

Bei Erstellung oder Pflege des Priorisierten Produkt Backlogs werden die priorisierten User Storys des bereits existierenden Priorisierten Produkt Backlogs und die priorisierte Risikenliste kombiniert, um ein aktualisiertes Priorisiertes Produkt Backlog zu erschaffen, welches alle identifizierten Risiken beinhaltet:

Schritte für das Aktualisieren des Priorisierten Produkt Backlogs mit identifizierten Risiken:

1. Erstellung einer priorisierten Risikoliste. (die Risiken können beispielsweise unter Verwendung der Methode des Erwarteten Geldwertes priorisiert werden)
2. Diejenigen identifizierten Risiken, die gemindert werden können, und bei denen das Team sich entscheidet, sie durch eine gezielte Risikomaßnahme während des Sprints zu mindern, werden ausgewählt.
3. Die in Schritt 2 identifizierten Risiken, die gemindert werden können, werden nach ihrem Wert priorisiert und dementsprechend dem Priorisierten Produkt Backlog hinzugefügt (als User Storys), um zu einem aktualisierten Priorisierten Produkt Backlog zu gelangen.

Abbildung 7-4 verdeutlicht den Ablauf der Risikopriorisierung.

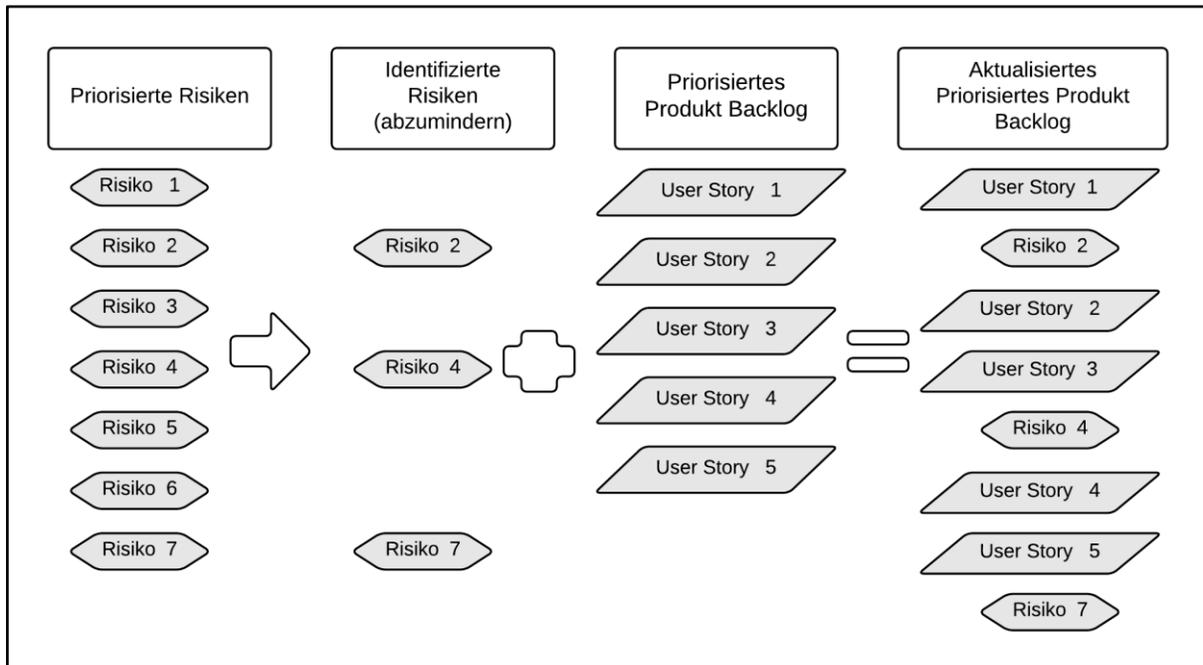


Abbildung 7-4: Ablauf der Risikopriorisierung

7.4.4 Risikominderung

Die Reaktion auf jedes Risiko wird von der Wahrscheinlichkeit und den Auswirkungen des Risikos abhängen. Jedoch ermöglicht die iterative Eigenschaft von Scrum mit seinen schnellen Umsetzungszeiten und Feedback-Zyklen eine frühzeitige Erkennung von Misserfolgen; es hat daher praktisch ein bereits integriertes, naturgegebenes Risikominderungsverfahren.

Risiken können durch die Anwendung einer Anzahl von Maßnahmen gemindert werden. In den meisten Situationen sind Maßnahmen entweder proaktiv oder reaktiv. Im Falle eines Risikos kann ein Plan B formuliert werden, der als Absicherung für den Fall des Eintretens des Risiko verwendet werden kann – ein solcher Plan B ist eine reaktive Maßnahme. Mitunter werden Risiken in Kauf genommen, ein Beispiel für eine Verhaltensweise im Bezug auf ein Risiko, welche weder proaktiv noch reaktiv ist. Risiken können aus verschiedensten Gründen in Kauf genommen werden, beispielsweise wenn Wahrscheinlichkeit oder die Auswirkungen des Risikos zu gering sind, um eine Maßnahme erforderlich zu machen. Inkaufnahme kann darüber hinaus auch stattfinden, wenn die Sorge um sekundäre Risiken den Product Owner von der Ergreifung einer Maßnahme abhält. Der Einsatz, den der Product Owner zur Minderung von Wahrscheinlichkeit oder Auswirkungen des Risikos - oder von beiden - erbringt, ist ein Beispiel für eine proaktive Maßnahme zur Minderung von Risiken.

Sobald identifizierte Risiken in das Priorisierte Produkt Backlog aufgenommen wurden (Vergleiche Abbildung 7-4), wird eine Anzahl von Risiken im Verlaufe des Prozesses *Zu liefernde Ergebnisse erzeugen* gemindert,

während die in Verbindung mit den durch das Priorisierte Produkt Backlog definierten User Storys stehenden Aufgaben vervollständigt werden.

In Scrum liegt die Verantwortung in Bezug auf das geschäftliche Handhaben von Risiken eindeutig beim Product Owner, während das Scrum Team dafür verantwortlich ist, die Risikomaßnahmen im Verlaufe des Sprints zu implementieren. Der Scrum Guidance Body kann um Rat bei der Implementierung von Risikomaßnahmen gefragt werden, auch um zu ermitteln, ob diese mit den Richtlinien der Organisation im Ganzen übereinstimmen. Der Scrum Master überwacht potentielle Risiken, die ein Projekt beeinflussen können, und informiert den Product Owner und das Scrum Team hierüber.

7.4.4.1 Risikobasierte Spikes

Ein Konzept, welches für das Identifizieren von Risiken nützlich sein kann, ist das der Risikobasierten Spikes. Spikes sind im Grunde ein Experiment, welches Forschung und Prototypen verwendet, um potenzielle Risiken besser zu verstehen. Bei einem Spike wird eine intensive zwei- bis dreitägige Übung durchgeführt (vorzugsweise zu Beginn eines Sprints, bevor Epic(s) entwickelt werden oder das priorisierte Produkt Backlog erstellt wird.), um dem Team bei der Bestimmung von Ungewissheiten, welche das Projekt beeinflussen könnten, zu helfen. Risikobasierte Spikes sind nützlich, wenn das Scrum Team mit neuen Technologien oder Methoden arbeitet und sich noch an diese gewöhnt, oder wenn die User Storys umfangreich sind. Darüber hinaus helfen sie dabei, Zeit und Aufwand genauer einzuschätzen.

Risikobasierte Spikes können auch während der gesamten Projektdauer durchgeführt und in jeden Sprint integriert werden. Solche Spikes müssen dem priorisierten Product Backlog hinzugefügt werden. Untersuchungen mittels risikobasierter Spikes werden verwendet, um künftige potenzielle Bedrohungen zu verringern.

7.4.5 Risikokommunikation

Da die Business Stakeholdern ein Interesse am Projekt haben, ist es wichtig, mit ihnen im Bezug auf Risiken zu kommunizieren. Die Informationen, welche den Business Stakeholdern im Bezug auf Risiken mitgeteilt werden, sollten die potentiellen Auswirkungen und die geplanten Maßnahmen für das jeweilige Risiko beinhalten. Diese Kommunikation findet permanent statt und sollte parallel zu den bisher genannten aufeinanderfolgenden vier Schritten - Risikoidentifikation, -bewertung, -priorisierung und -minderung - durchgeführt werden. Das Scrum Team kann darüber hinaus spezifische, mit seinen Aufgaben verknüpfte Risiken mit dem Scrum Master während des Täglichen Standup Meetings besprechen. Der Product Owner ist für die Priorisierung von Risiken und die Verteilung der priorisierten Liste an das Scrum Team verantwortlich.

Ein wichtiges Werkzeug, welches zur Weitervermittlung von Informationen im Bezug auf Risiken genutzt werden kann, ist das Risiko-Burndown-Chart.

7.4.5.1 Risiko-Burndown-Chart

Risikomanagement ist unabdingbar zur Absicherung von Werterschaffung; daher werden Maßnahmen des Risikomanagements während der gesamten Aktivphase des Projekts und nicht nur bei Beginn des Projekts durchgeführt.

Jedes Risiko kann mit Hilfe von unterschiedlichen Hilfsmitteln der Risikobewertung ausgewertet werden. Dennoch ist das bevorzugte Mittel zur Bewertung von Risiken bei der Erstellung eines Risiko-Burndown-Charts der in Abschnitt 7.4.2.1 beschriebene Erwartete Geldwert (EMV).

Die gesammelten Informationen im Verlaufe der Risikobewertung können zur Erstellung eines Risiko-Burndown-Charts genutzt werden. Dieses stellt die Gesamtheit der Schwere des Risikos des Projekts im Verlaufe der Zeit dar. Die Wahrscheinlichkeiten von verschiedenen Risiken werden zusammengenommen, um das kumulierte Risiko auf der y-Axe darzustellen. Die erstmalige Identifikation und Bewertung von Risiken für das Projekt und die Erstellung des Risiko-Burndown-Charts werden zu Beginn durchgeführt.

Dann sollte in vorherbestimmten Zeitabständen nach neuen Risiken gesucht werden, um diese zu bewerten und die verbliebenen Risiken neu zu bewerten, so dass diese entsprechend im Chart aktualisiert werden können. Der angemessene Zeitpunkt für die Durchführung dieser Aktualisierungen ist das Sprint-Planungs-Meeting. Die Nachverfolgung von Risiken auf diese Weise ist, was es dem Team ermöglicht, Tendenzen bei der Risikobelastung zu erkennen und gegebenenfalls angemessene Maßnahmen zu ergreifen.

Abbildung 7-5 zeigt ein Beispiel für einen Risiko-Burndown-Chart.

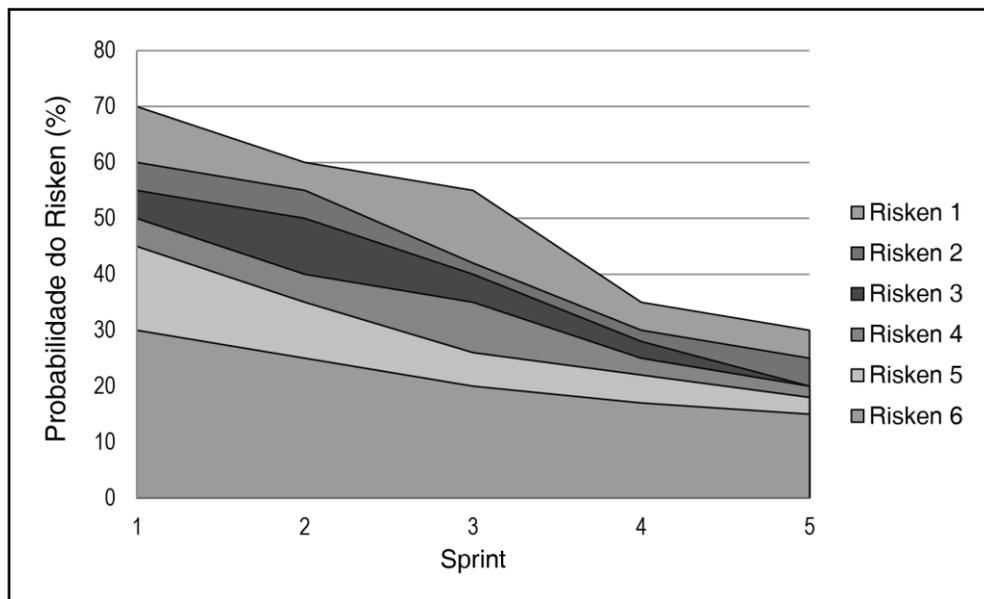


Abbildung 7-5: Beispiel eines Risiko-Burndown-Charts

7.5 Minimierung von Risiken mit Hilfe von Scrum

Als iterativer Vorgang im Stile von Agile verringert der Scrum-Rahmen von Natur aus Risiken. Die folgenden Scrum-Methoden erleichtern das effektive Handhaben von Risiken:

1. Flexibilität verringert Risiken, welche mit der Geschäftsumgebung verbunden sind

Risiken werden im Rahmen von Scrum durch die Flexibilität beim Hinzufügen und Modifizieren von Anforderungen zu jeder Zeit der Aktivphase des Projekts weitgehend minimiert. Dies ermöglicht es der Organisation, auf Gefahren oder Gelegenheiten des geschäftlichen Umfeldes und unvorhergesehene Anforderungen zu reagieren, wenn diese entstehen, mit zumeist geringen Kosten für das Handhaben solcher Risiken.

2. Regelmäßiges Feedback reduziert erwartungsbezogene Risiken

Aufgrund seiner iterativen Eigenschaft sorgt der Scrum Rahmen für eine große Anzahl von Gelegenheiten, Feedback zu erhalten und Erwartungen durch die gesamte Aktivphase des Projektes hindurch festzusetzen. Dies sorgt dafür, dass sowohl die Business Stakeholdern des Projekts, als auch das Team, nicht durch falsch kommunizierte Erwartungen auf dem falschen Fuß erwischt werden.

3. Team-Verantwortung verringert Abschätzungsrisiken

Das Scrum Team kalkuliert und übernimmt die Verantwortung für Bestandteile des Sprint Backlogs, was zu einer akkurateren Abschätzung und zeitgemäßen Lieferung der Produktstufen führt.

4. Transparenz verringert das Nicht-Entdecken von Risiken

Der Scrum Grundsatz der Transparenz, auf welchem der Rahmen aufbaut, sorgt dafür, dass Risiken früh erkannt werden, was zu einer besseren Handhabung und Minderung von Risiken führt. Darüber hinaus können bei einem Scrum der Scrums Meeting Impedimente, welche ein Team zu diesem Zeitpunkt beschäftigen, als zukünftiges Risiko für andere Teams benannt werden, was wiederum in ein aktualisiertes Impediment Log aufgenommen werden sollte.

5. Iterative Lieferung reduziert Investment-Risiken

Das kontinuierliche Liefern von Werten im Verlaufe der Aktivphase eines Scrumprojektes, bei dem nach jedem Sprint potenziell versandfähige zu liefernde Ergebnisse erzeugt werden, verringert das Investment-Risiko für den Kunden.

7.6 Risiken bei Portfolios und Programmen

Während einige Risiken speziell mit einem individuellen Projekt verbunden sind, entstammen andere Programmen oder Portfolios und werden üblicherweise auch dort gehandhabt. Jedoch beeinflussen Risiken im Rahmen eines Portfolios oder Programms auch diejenigen Projekte, die Teil des jeweiligen Portfolios oder Programms sind. Während der Risikobewertung in Portfolios und Programmen müssen relevante Informationen über ein Risiko dem Product Owner und dem Scrum Team mitgeteilt werden, wenn dieses Risiko ein individuelles Projekt berührt.

Wenn das Programm oder Portfolio Team ein Risiko mit Auswirkungen auf ein individuelles Projekt meldet, kann es - abhängig von Schwere und Priorität - sein, dass das Scrum Team den gegenwärtigen Sprint abbrechen und neu planen muss, um das Risiko anzugehen. Bei weniger dringenden Risiken kann das Team den gegenwärtigen Sprint fortsetzen und das Risiko in einem späteren Sprint behandeln.

7.6.1 Bei Portfolios

1. Wenn Risiken bei einem Portfolio identifiziert werden, muss der Product Owner des Portfolios diese festhalten und die zeitliche Nähe, Wahrscheinlichkeit und Auswirkungen jedes identifizierten Risikos bewerten, um dieses zu priorisieren und angemessene Maßnahmen für das Portfolio zu ermitteln.
2. Der Product Owner des Portfolios wird darüber hinaus die Risiken den relevanten Business Stakeholdern, den Programm-Teams, und den Projekt-Teams mitteilen müssen. In manchen Fällen, kann das Portfolioteam dazu gezwungen sein, die Verantwortung für eines spezifisches Risiko zu übernehmen.

7.6.2 Bei Programmen

1. Wenn Risiken bei Programmen identifiziert werden, sollte der Product Owner des Programms diese in ein risikobereinigtes priorisiertes Produkt Backlog des Programms integrieren und die zeitliche Nähe, Wahrscheinlichkeit und Auswirkungen jedes identifizierten Risikos bewerten, um dieses zu priorisieren und angemessene Maßnahmen auf einem Programmlevel zu finden.
2. Der Product Owner des Programms wird darüber hinaus die Risiken den relevanten Business Stakeholdern und den Projektteams mitteilen müssen. In manchen Fällen, muss das Programmteam die Verantwortung für eines spezifisches Risiko übernehmen.

Abbildung 7-6 veranschaulicht, wie Risiken im Scrumfluss für Portfolios und auch für Programme gehandhabt werden können.

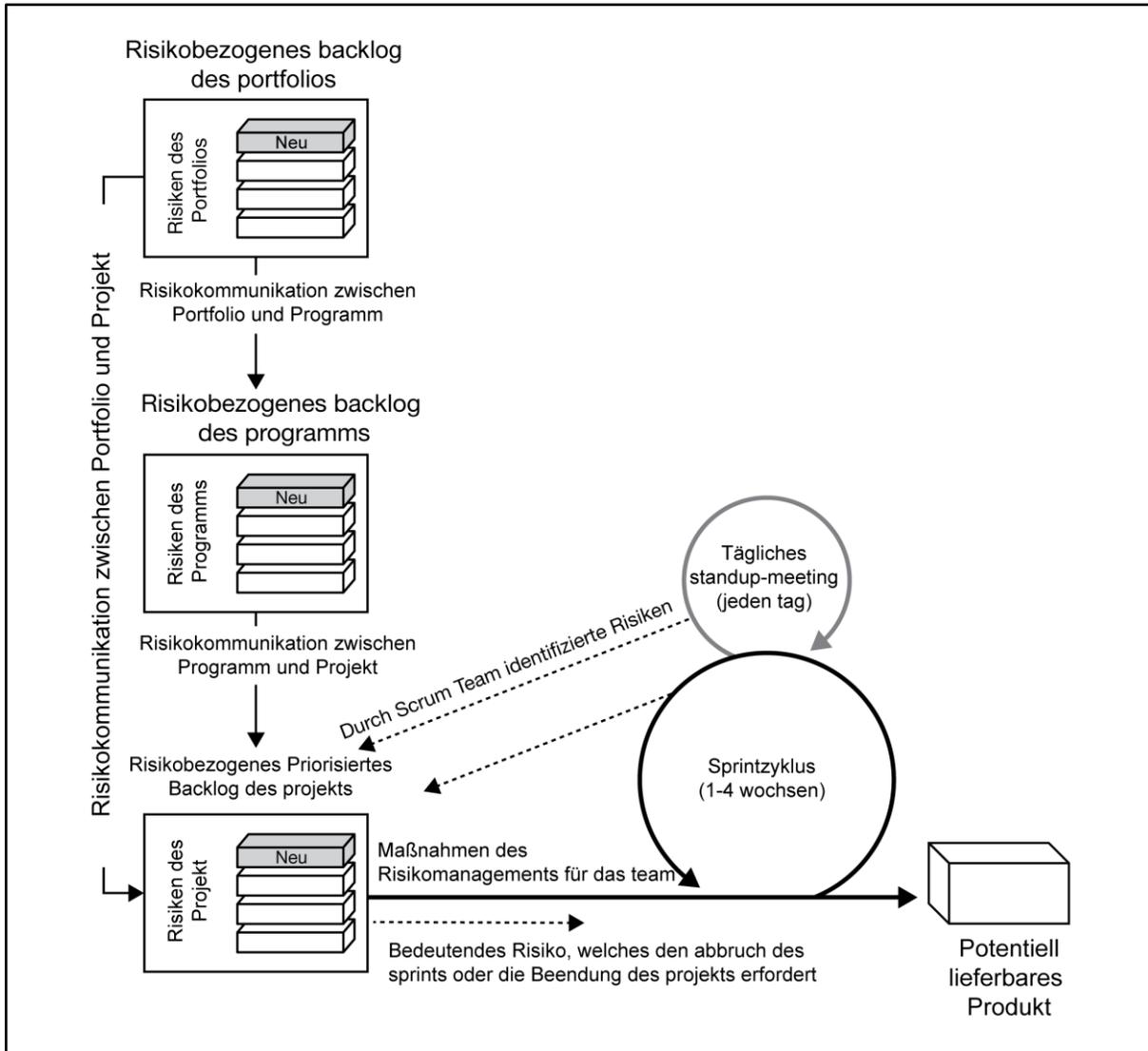


Abbildung 7-6: Handhabung von Risiken in Portfolios und Programmen

7.7 Zusammenfassung der Verantwortlichkeiten

In Scrum werden die Aktivitäten des Risikomanagements zwischen den verschiedenen Rollen aufgeteilt, so dass sowohl der den Vorgang leitende Scrum Master als auch jeder im Scrum Team einen Teil der Verantwortung übernimmt.

Rolle	Verantwortlichkeiten
Scrum Guidance Body	<ul style="list-style-type: none"> • Sorgt für allgemeine Orientierungshilfe für den Vorgang des Risikomanagements, der während des gesamten Projekts zu befolgen ist.
Product Owner des Portfolios	<ul style="list-style-type: none"> • Hält Risiken auf Portfoliolevel fest und bewertet diese. • Priorisiert Risiken und teilt diese den relevanten Business Stakeholdern, Programm-Teams und Projekt-Teams mit.
Scrum Master des Portfolios	<ul style="list-style-type: none"> • Leitet die Identifikation, Bewertung und Kommunikation von Risiken auf dem Portfoliolevel.
Product Owner des Programms	<ul style="list-style-type: none"> • Hält Risiken auf Programmlevel fest und bewertet diese • Priorisiert Risiken und teilt diese den relevanten Business Stakeholdern, und den Projekt-Teams mit.
Scrum Master des Programms	<ul style="list-style-type: none"> • Leitet die Identifikation, Bewertung und Kommunikation von Risiken auf dem Programmlevel.
Business Stakeholdern	<ul style="list-style-type: none"> • Steht mit dem Scrum Kernteam in Verbindung, um dieses mit Vorgaben für das Handhaben von Risiken, welche das Erreichen der erwarteten Ergebnisse berühren, zu versorgen, und profitiert vom Projekt.
Product Owner/Chief Product Owner	<ul style="list-style-type: none"> • Hält Risiken für das Projekt fest und bewertet diese. • Priorisiert Risiken und teilt diese den relevanten Business Stakeholdern, Programm-Teams und Portfolio-Teams mit. • Sorgt dafür, dass die Risikolevel des Projekts im akzeptablen Rahmen bleiben.
Scrum Master/Chief Scrum Master	<ul style="list-style-type: none"> • Leitet die Identifikation, Bewertung und Kommunikation von Risiken durch das Scrum Team.
Scrum Team	<ul style="list-style-type: none"> • Identifiziert Risiken während der Entwicklung des Produkts und bei der Erzeugung der zu liefernden Ergebnisse. • Implementiert Maßnahmen des Risikomanagements nach den Vorgaben des Product Owners.

Tabelle 7-1: Zusammenfassung der Verantwortlichkeiten im Bezug auf Risiken

7.8 Scrum versus traditionelles Projektmanagement

Scrum und die meisten traditionellen Projektmanagement-Methoden definieren Risiko als 'unsichere Ereignisse, welche das Erreichen von Projektzielen positiv oder negativ beeinflussen kann.' Auch werden Risiken identifiziert, bewertet, eingeplant und beständig kommuniziert.

Traditionelle Projektmanagement-Methoden betonen eine detaillierte vorhergehende Planung zur Identifikation, Bewertung und Entscheidung von Maßnahmen für alle Projekt-Risiken. Im Laufe der Durchführung des Projekts kann jedes Teammitglied Risiken identifizieren, die dann durch den Projektmanager oder das Projektmanagement-Büro (PMO) / das Betreuungspersonal des Projekts in das Risiko Log/Register übertragen werden können. Der Projektmanager überwacht und kontrolliert regelmäßig alle Risiken, und wählt für gewöhnlich spezifische Individuen innerhalb des Teams aus, um ihnen die Verantwortung für unterschiedliche Aspekte von Risiken zuzuweisen.

Bei Scrum kann jedes Mitglied des Scrum Teams Risiken identifizieren und der Product Owner kann diese identifizierten Risiken in das risikobereinigte Priorisierte Produkt Backlog übernehmen. Die Scrum Prinzipien der Empirischen Prozesssteuerung und der iterativen Entwicklung ermöglichen es dem Scrum Team, die Risiken beständig zu identifizieren und in das Priorisierte Produkt Backlog aufzunehmen, wo solche Risiken zusammen mit anderen existierenden User Storys innerhalb des Backlogs priorisiert werden, um bei den folgenden Sprints abgemildert zu werden. Das Scrum Team hat die kollektive Verantwortung für die Handhabung aller Risiken während eines Sprints.

8. INITIIERUNG

Dieses Kapitel beinhaltet die Prozesse, welche mit dem Beginn eines Projekts verbunden sind: *Projektvision erstellen, Scrum Master und Business Stakeholdern identifizieren, Scrum Team bilden, Epic(s) entwickeln, Priorisiertes Produkt Backlog erstellen, und Release Planung durchführen.*

Die Initiierung ist nach der Definition des *Scrum Body of Knowledge (SBOK®)* auf folgendes anwendbar:

- Portfolios, Programme, und/oder Projekte in jeder Branche.
- Produkte, Dienstleistungen, oder jegliche an Business Stakeholdern zu liefernde Ergebnisse Projekte jeglicher Größe und Komplexität

Der Begriff "Produkt" in dieser Wissenssammlung kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann auf jedes Projekt in jeder Branche effektiv angewendet werden — von kleinen Projekten oder Teams mit lediglich sechs Team-Mitgliedern bis hin zu großen, komplexen Projekten mit mehreren Hundert Team-Mitgliedern.

Um die beste Anwendung des Scrum Rahmens zu ermöglichen, kennzeichnet dieses Kapitel Inputs, Hilfsmittel und Outputs für jeden Prozess entweder als „verbindlich“ oder „optional“. Inputs, Hilfsmittel und Outputs, die mit einem Sternchen (*) gekennzeichnet sind, sind verbindlich oder werden als kritisch fuer den Erfolg des Projekts angesehen, wohingegen jene ohne Sternchen optional sind.

Es wird empfohlen, dass das Scrum Team und diejenigen, welche sich erst mit dem Scrum Rahmen vertraut machen sich primär mit den verbindlichen Inputs, Hilfsmittel und Outputs beschäftigen, während Product Owner, Scrum Master, und andere Scrum Praktiker mit größerer Erfahrung danach streben sollten, sich ein noch tieferes Wissen über die Informationen in diesem gesamten Kapitel anzueignen

Dieses Kapitel ist aus der Perspektive eines einzelnen Scrum Teams geschrieben worden, welches an einem Sprint arbeitet, um potenziell versandfähige Ergebnisse als Teil eines größeren Projektes, eines Programms oder eines Portfolios zu produzieren. Weitere Informationen, die sich auf den Gebrauch von Scrum für große Projekte beziehen, sind in Kapitel 13 zu finden, die Skalierung von Scrum für Programme und Portfolios befinden sich in Kapitel 14.

Die Initiierungs-Phase steht am Anfang eines Scrum-Projekts. In dieser Phase werden das Scrum Kernte Team und die Business Stakeholdern identifiziert, beginnend mit dem Product Owner, der eine Projektvision erstellt, die als Leitfaden für das gesamte Projekt dient. Auf der Grundlage der Projektvision wird ein erster Satz von Anforderungen ermittelt und in Form von Epics dokumentiert.

Diese ersten Anforderungen werden priorisiert und dazu verwendet, ein erstes priorisiertes Produkt Backlog zu erstellen (dies ist das Dokument, in dem alle Anforderungen eines Scrum-Projekts festgehalten werden). Im letzten Schritt der Initiierungs-Phase wird eine Release Planung für das Gesamtprojekt erstellt. In der Initiierungs-Phase wird kein umfassender und detaillierter Plan für das gesamte Projekt erstellt. Es besteht keine Notwendigkeit für einen umfassenden Plan, da Änderungen erwartet werden und aufgrund des iterativen Prinzips, das in den Scrum-Prozessen verankert ist, leicht in ein Scrum-Projekt integriert werden können.

Das Ziel der Initiierungs-Phase eines Scrum-Projekts besteht vielmehr darin, einen guten Anfangsplan für das Projekt zu erstellen, der sich an den geschäftlichen Erfordernissen und/oder an hoch-prioren Vorschriften orientiert. Diese Phase sollte kurz sein, damit die Wertschöpfung im Projekt so schnell wie möglich beginnen kann.

Darüber hinaus ist es ebenfalls von Bedeutung zu verstehen, dass, obwohl alle Vorgänge einzeln im SBOK® definiert sind, diese nicht notwendigerweise auch nacheinander oder voneinander getrennt ablaufen. Mitunter ist es nützlicher, einige Vorgänge, je nach den speziellen Bedürfnissen eines jeden Projekts, miteinander zu kombinieren.

Abbildung 8-1 gibt einen Überblick über die Prozesse der Initiierungsphase, welche nachstehend gelistet sind:

8.1 Projektvision erstellen—In diesem Prozess wird der Product Owner identifiziert. Der Product Owner erstellt dann auf Basis des Business Cases für das Projekt eine Projektvision, die als Inspiration und Richtschnur für das gesamte Projekt dient.

8.2 Scrum Master und Business Stakeholdern identifizieren— In diesem Prozess wird der Scrum Master anhand spezifischer Auswahlkriterien bestimmt, mit denen die für diese wichtige Rolle erforderlichen Soft Skills und Scrum-Kenntnisse effektiv bewertet werden können. Darüber hinaus werden während dieses Prozesses auch die Business Stakeholdern ermittelt.

8.3 Scrum Team bilden— In diesem Prozess werden die Mitglieder des Scrum-Teams auf der Grundlage der Fähigkeiten ermittelt, die für die Fertigstellung der Projektleistungen erforderlich sind, sowie unter Berücksichtigung der Verfügbarkeit, der Kosten und der Soft Skills, die für die Mitglieder eines Scrum-Teams wichtig sind. Üblicherweise ist der Product Owner für die Auswahl der Teammitglieder verantwortlich, doch oft geschieht dies in Zusammenarbeit mit dem Scrum Master.

8.4 Epic(s) entwickeln—In diesem Prozess dient die Projektvisionserklärung als Grundlage für die Entwicklung von Epics, die die High-Level-Anforderungen für das Projekt definieren. Der Product Owner kann User Group Meetings und andere Tools nutzen, um die Anforderungen der Business Stakeholdern zu sammeln.

8.5 Priorisiertes Produkt Backlog erstellen— In diesem Prozess werden die Epics verfeinert, ausgearbeitet und, was am wichtigsten ist, nach ihrem jeweiligen Geschäftswert priorisiert, um ein priorisiertes Product Backlog für das Projekt zu erstellen. Zusätzlich legen der Product Owner und das Scrum Team auf der Grundlage der Empfehlungen des Scrum Guidance Body die Done Kriterien für das Projekt fest.

8.6 Release-Planung durchführen— In diesem Prozess entwickelt der Product Owner mit Unterstützung des Scrum-Teams die anfängliche Release-Planung, die den Business Stakeholdern mitgeteilt und mit ihnen geteilt wird. Es wird davon ausgegangen, dass die iterative Natur von Scrum zukünftige Anpassungen des Release Plans erforderlich machen kann. Die Länge der einzelnen Sprints wird ebenfalls in diesem Prozess festgelegt.

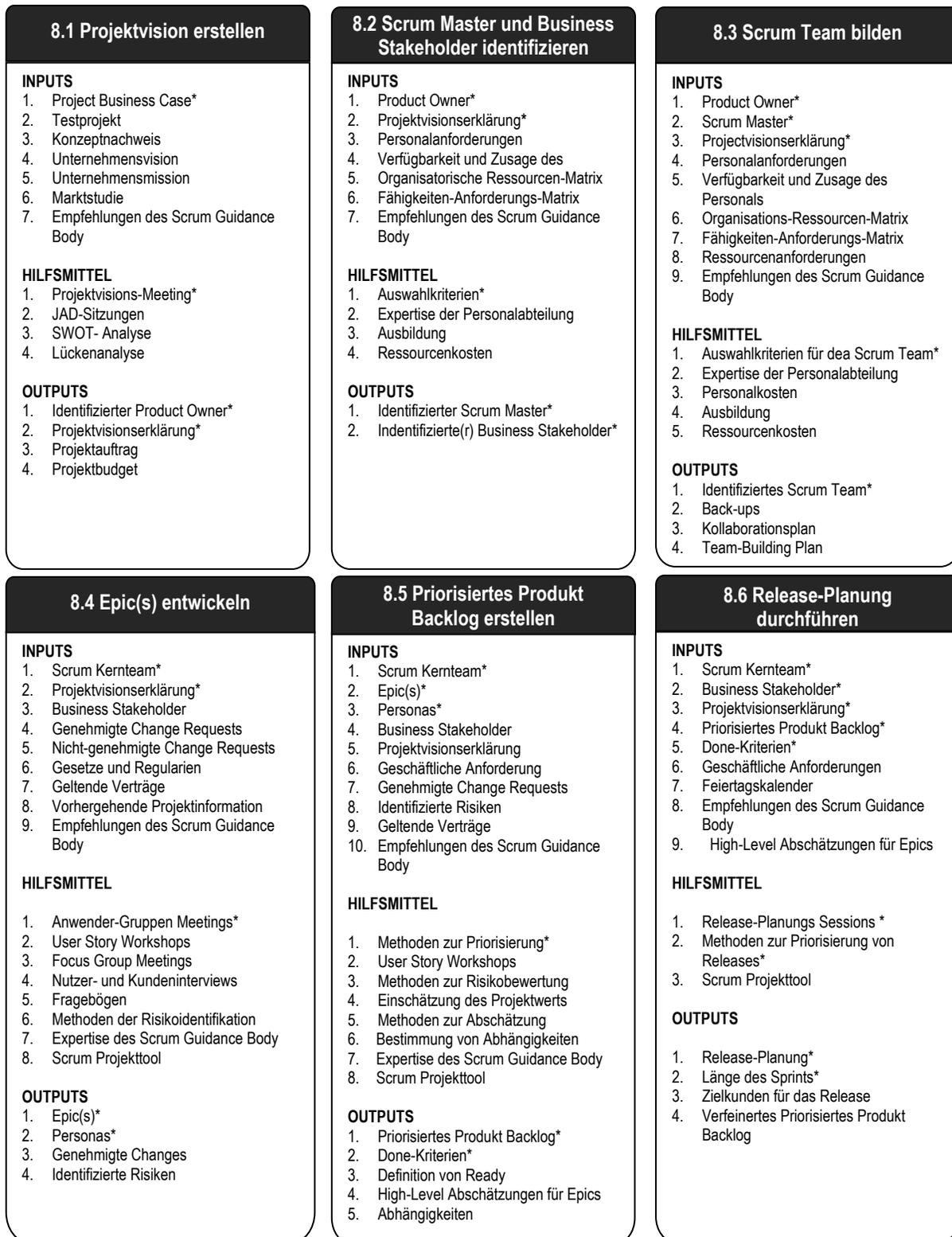


Abbildung 8-1: Überblick über die Initiierungsphase

Abbildung 8-2 unten zeigt die verbindlichen Inputs, Hilfsmittel und Outputs für Prozesse in Initiierungsphase.

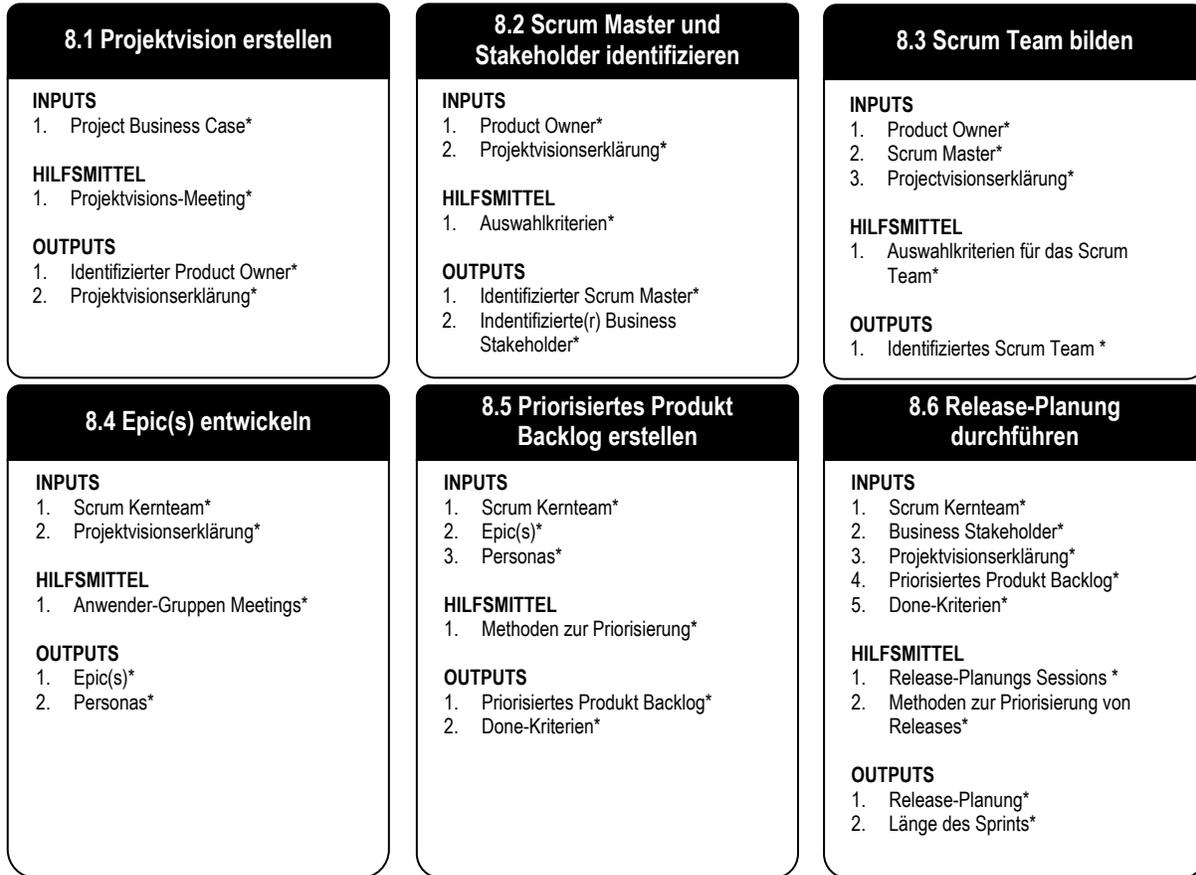


Abbildung 8-2: Überblick über die Initiierungsphase (Kernpunkte)

8.1 Projektvision Erstellen

8.1 Projektvision erstellen—In diesem Prozess wird der Product Owner identifiziert. Der Product Owner erstellt dann auf Basis des Business Cases für das Projekt eine Projektvision, die als Inspiration und Richtschnur für das gesamte Projekt dient.

Abbildung 8-3 zeigt um die Inputs, Hilfsmittel und Outputs für *Projektvision Erstellen* Prozess.

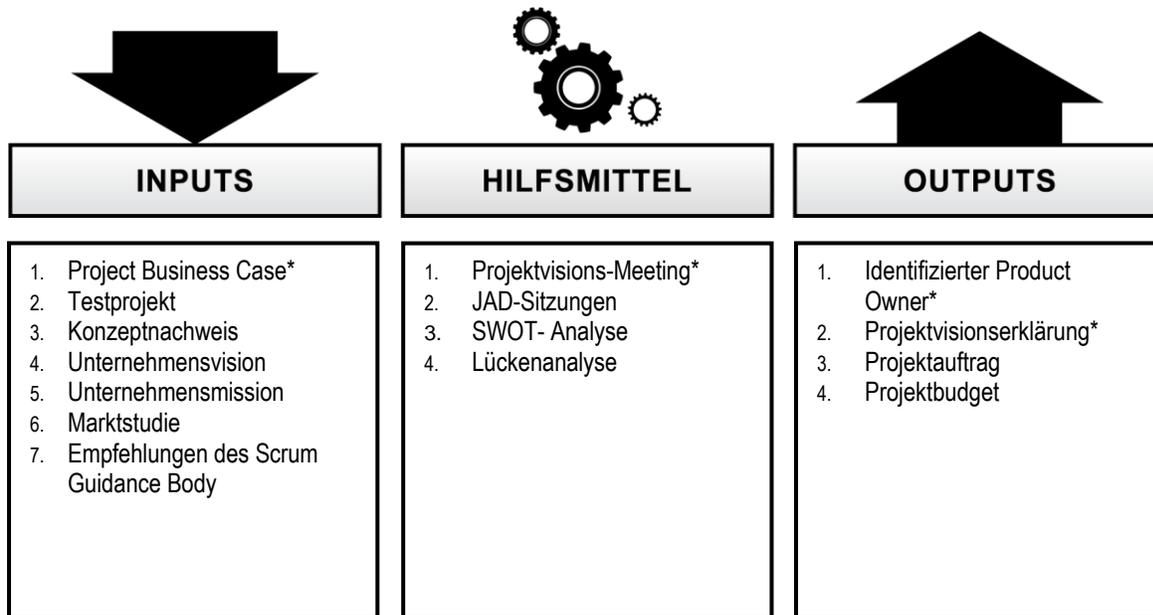


Abbildung 8-3: Projektvision erstellen—Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

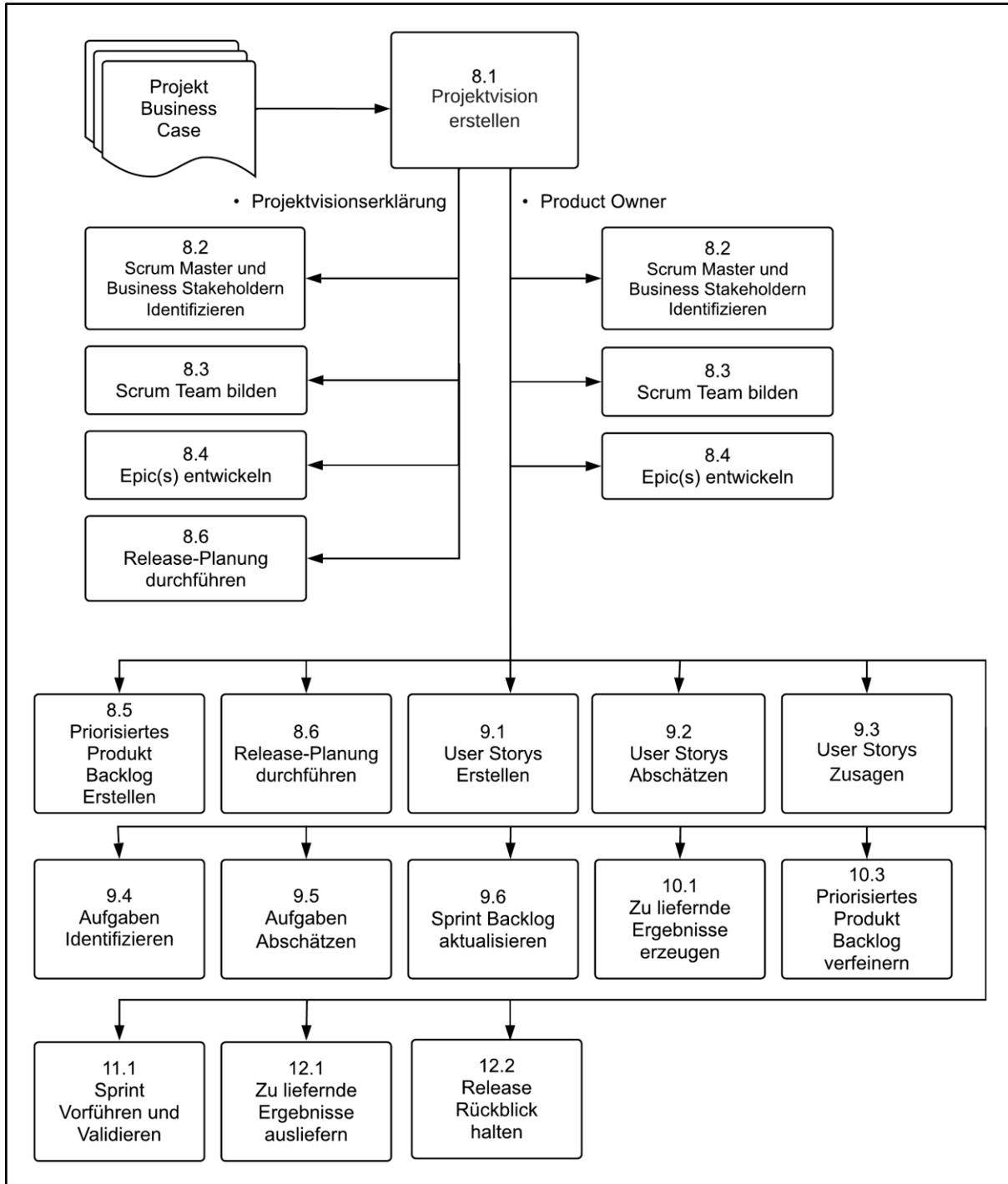


Abbildung 8-4: Projektvision erstellen–Datenflussdiagramm

8.1.1 Inputs

8.1.1.1 Projekt Business Case*

Das Projekt beginnt mit der Vorstellung des Projekt Business Cases vor den Business Stakeholdern und den Sponsoren. Die Business Stakeholder verstehen die zu erwartenden Geschäftsvorteile des Projekts, und die Sponsoren bestätigen, dass sie die finanziellen Ressourcen für das Projekt bereitstellen werden.

Ein Business Case kann sowohl ein gut strukturiertes Dokument als auch einfach eine mündliche Aussage sein, wenn diese die Gründe für die Initiierung des Projektes ausdrücken. Er kann sowohl formal und umfangreich als auch formlos und kurz gehalten sein. Unabhängig vom Format enthält er oft wesentliche Informationen über den Hintergrund des Projekts, die beabsichtigten Geschäftsziele und die gewünschten Ergebnisse, einen Bericht über die SWOT- und Lückenanalyse, eine Liste der identifizierten Risiken und Schätzungen zu Zeit, Aufwand und Kosten. Der Business Case für das Projekt sollte auch alle relevanten externen Faktoren berücksichtigen, z. B. gesetzliche Vorschriften, behördliche Auflagen, Datenschutzbelange und so weiter.

8

8.1.1.2 Testprojekt

Falls möglich, kann eine kleinformatige Demonstration oder ein Testprojekt als Experiment durchgeführt werden, um Funktionsfähigkeit, Zeit, Kosten, Risiken und andere mögliche Ergebnisse des eigentlichen Projekts vorherzusagen und einzuschätzen. Dies hilft dabei, das praktische Umfeld zu bewerten und die eigentliche Projektentwicklung vor der Initiierung des vollen Projektes anzuleiten.

8.1.1.3 Konzeptnachweis

Ein Konzeptnachweis demonstriert und verifiziert, dass die Idee hinter dem gegenwärtigen Projekt potentiell in der realen Welt realisierbar ist. Er ist, oft in Form eines Prototyps, daraufhin entwickelt worden, die finanzielle und technische Realisierbarkeit festzustellen, beim Verstehen von Anforderungen zu helfen, und die Zuordnung von Entwicklungsentscheidungen in der Frühphase des Projekts zu unterstützen. Allerdings muss der Konzeptnachweis nicht zwingend die eigentlichen zu liefernden Ergebnisse des Projektes repräsentieren.

8.1.1.4 Unternehmensvision

Das Verstehen der Unternehmensvision hilft dem Projekt, seinen Fokus auf die Ziele der Organisation und das zukünftige Potential des Unternehmens zu richten. Der Product Owner kann aus der Unternehmensvision Richtlinien und Anweisungen entnehmen und daraus die Projektvisionserklärung entwickeln.

8.1.1.5 Unternehmensmission

Die Unternehmensmission stellt einen Rahmen für die Formulierung der Unternehmensstrategien bereit und leitet den gesamten Entscheidungsfindungsprozess des Unternehmens an. Die Projektvision muss so ausgearbeitet werden, dass ihre Erfüllung der Organisation bei der Erfüllung ihrer Mission hilft.

8.1.1.6 Marktstudie

Marktstudie bezieht sich auf organisierte(n) Erforschung, Sammlung, Abgleich und Analyse von Daten, welche mit den Kundenwünschen für ein Produkt zusammenhängen. Sie beinhaltet oft ausführliche Daten über Markttrends, Marktaufteilung, und Marketing Abläufe. Eine Marktstudie kann darüber hinaus auch eine analytische Studie der Konkurrenten beinhalten, welche ein besseres Verständnis für die Stärken und Schwächen der Konkurrenten ermöglicht, und den Entscheidungsträgern bei der Entwicklung von besser positionierten Produkten helfen kann.

8.1.1.7 Empfehlungen des Scrum Guidance Body

Der Scrum Guidance Body (SGB) ist eine ist eine optionale Rolle. Er besteht üblicherweise aus einer Gruppe von Dokumenten und/oder einer Gruppe von Experten, welche typischerweise an der Definition von Zielen im Bezug auf Qualität, staatliche Vorgaben, Sicherheit und andere organisatorische Schlüsselparameter beteiligt sind. Diese Ziele stellen Richtlinien für die Arbeit des Product Owners, des Scrum Masters und des Scrum Teams bereit. Der Scrum Guidance Body kann auch dabei helfen, bewährte Verfahren zu sammeln und darzustellen, welche bei allen Scrum Projekten der Organisation genutzt werden sollten.

Der Scrum Guidance Body trifft keine Entscheidungen in Bezug auf das Projekt. Stattdessen fungiert er als ratgebende oder leitende Struktur für alle Hierarchie-Ebenen der Projektorganisation - das Portfolio, das Programm und das Projekt. Scrum Teams haben die Möglichkeit, den Scrum Guidance Body wenn nötig um Rat zu fragen.

Es ist wichtig, sicherzustellen, dass die Projektvision sich mit den Empfehlungen deckt, welche von Scrum Guidance Body bereitgestellt werden, und dass die Prozesse allen Standards und Richtlinien entsprechen, die vom Scrum Guidance Body festgelegt werden.

8.1.2 Hilfsmittel

8.1.2.1 Projektvisions-Meeting*

Ein Projektvisions-Meeting ist ein Treffen der Programm Business Stakeholdern, des Programm Product Owners, und des Programm Scrum Masters. Es hilft bei der Identifizierung von Geschäftsumfeld, von geschäftlichen Anforderungen und den Erwartungen der Business Stakeholdern, um eine effektive Projektvisionserklärung zu entwickeln. Scrum baut auf enge Einbindung und Zusammenarbeit aller geschäftlichen Repräsentanten, um ihre Zustimmung zum Projekt zu erhalten und einen größeren Wert zu liefern.

8.1.2.2 JAD Sitzungen

Eine Joint Application Design (JAD)-Sitzung ist eine Methode zur Sammlung von Anforderungen. Sie ist ein sehr strukturierter moderierter Workshop, der das Erstellen einer Projektvision beschleunigt, in dem es den Business Stakeholdern und anderen Entscheidungsträgern ermöglicht, im Bezug auf den Umfang, die Ziele und andere Aspekte eines Projektes zu einem Konsens zu gelangen.

Die Sitzung besteht aus Methoden zur Verbesserung der Anwenderbeteiligung, Beschleunigung der Entwicklung und Verbesserung der Spezifikationen. Relevante Programm Business Stakeholdern, Programm Product Owner, Program Scrum Master und Chief Product Owner können sich treffen, um die erwünschten Geschäftsergebnisse zu umreißen und zu analysieren und ihre Vision für das Scrum Projekt zu verdeutlichen.

8.1.2.3 SWOT-Analyse

SWOT ist eine strukturierte Methode der Projektplanung, welche dabei hilft, die Stärken, Schwächen, Gelegenheiten und Gefahren im Bezug auf ein Projekt zu bewerten. Diese Art der Analyse hilft bei der Identifizierung sowohl von internen als auch externen Faktoren, die sich auf das Projekt auswirken können. Stärken und Schwächen sind interne Faktoren, Gelegenheiten und Gefahren externe Faktoren. Das Identifizieren dieser Faktoren hilft den Business Stakeholdern und Entscheidungsträgern die Prozesse, Hilfsmittel und Methoden festzulegen, welche zum Erreichen der Projektziele angewandt werden. Das Durchführen einer SWOT-Analyse ermöglicht die frühe Identifikation von Prioritäten, möglichen Veränderungen, und Risiken.

8.1.2.4 Lückenanalyse

Die Lückenanalyse ist eine Methode zum Vergleich des gegenwärtigen, tatsächlichen Zustands mit einem gewünschten Zustand. Bei einer Organisation umfasst dies die Feststellung und Dokumentation der Differenz zwischen den gegenwärtigen Geschäftsfähigkeiten und den am Ende erwünschten Fähigkeiten. Ein Projekt wird üblicherweise initiiert, um die Organisation in einen gewünschten Zustand zu versetzen, sodass eine Lückenanalyse den Entscheidungsträgern dabei hilft, die Bedürfnisse für ein Projekt festzustellen.

Die Hauptschritte einer Lückenanalyse sind in Abbildung 8-5 wiedergegeben.

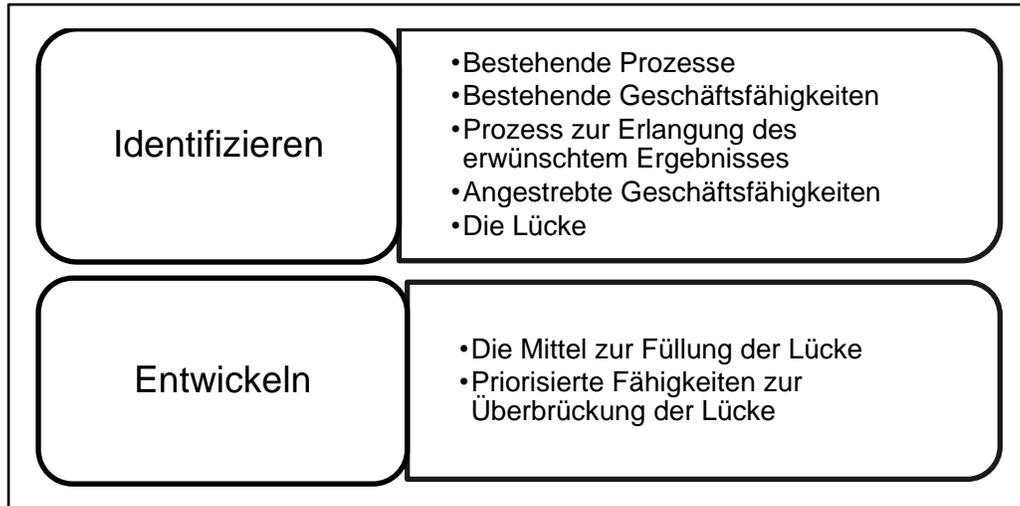


Abbildung 8-5: Der Vorgang der Lückenanalyse

8.1.3 Outputs

8.1.3.1 Identifizierter Product Owner*

Eines der Ergebnisse dieses Prozesses ist die Identifikation des Product Owners. Der Product Owner ist dafür verantwortlich, dass der maximale Geschäftswert des Projekts erreicht wird. Sie/er ist auch dafür verantwortlich, Kundenanforderungen zu formulieren und die geschäftliche Rechtfertigung des Programms zu erhalten. Der Product Owner vertritt die Stimme des Kunden.

Die Rolle des Product Owners ist detaillierter in Abschnitt 3.4 beschrieben.

8.1.3.2 Projektvisionserklärung*

Das Schlüsselergebnis des Prozesses *Projektvision erstellen* ist eine klar strukturierte Projektvisionserklärung. Eine gute Projektvision erklärt die Geschäftsbedürfnisse, die das Projekt befriedigen soll, anstatt zu beschreiben, wie es diese befriedigt.

Die Projektvisionserklärung sollte nicht allzu spezifisch sein und Raum für Flexibilität bieten. Es ist möglich, dass das gegenwärtige Verständnis des Projekts auf Annahmen basiert, welche sich im Verlaufe des Projektes ändern, weshalb es wichtig ist, dass die Projektvision flexibel genug ist, um diese Veränderungen zu umfassen. Die Projektvision sollte sich mit dem Problem und nicht mit der Lösung befassen.

8.1.3.3 Projektauftrag

Ein Projektauftrag ist eine offizielle Aussage zu erwünschten Zielen und Ergebnissen eines Projekts. In vielen Organisationen ist der Projektauftrag das Dokument, das das Projekt offiziell und förmlich autorisiert, und dem Team damit die schriftliche Genehmigung zum Beginn der Projektarbeit gibt.

8.1.3.4 Projektbudget

Das Projektbudget ist ein finanzielles Dokument, welches die Kosten für Menschen, Materialien, und andere damit verbundene Ausgaben bei einem Projekt beinhaltet. Das Projektbudget wird typischerweise von den Sponsoren abgesegnet, um sicherzustellen, dass genügend Gelder verfügbar sind. Nach der Absegnung fällt das regelmäßige Management des Projektbudgets an den Product Owner und den Scrum Master, um so abzusichern, dass Arbeitskräfte und andere Ressourcen, welche vom Projekt benötigt werden, verfügbar sind.

8.2 Scrum Master und Business Stakeholdern Identifizieren

In diesem Prozess wird der Scrum Master anhand spezifischer Auswahlkriterien bestimmt, mit denen die für diese wichtige Rolle erforderlichen Soft Skills und Scrum-Kenntnisse effektiv bewertet werden können. Darüber hinaus werden während dieses Prozesses auch die Business Stakeholdern ermittelt.

Abbildung 8-6 zeigt um so Inputs, Hilfsmittel und Outputs für *Scrum Master und Business Stakeholdern Identifizieren* Prozess.

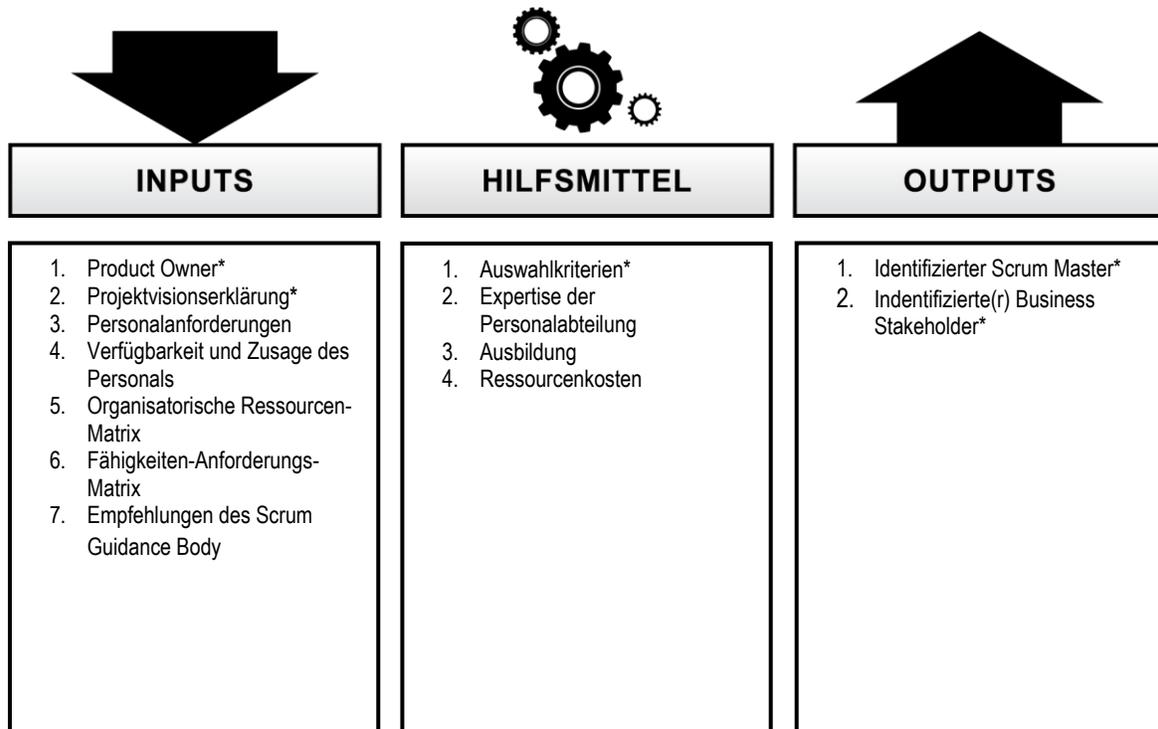


Abbildung 8-6: Scrum Master und Business Stakeholdern identifizieren—Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

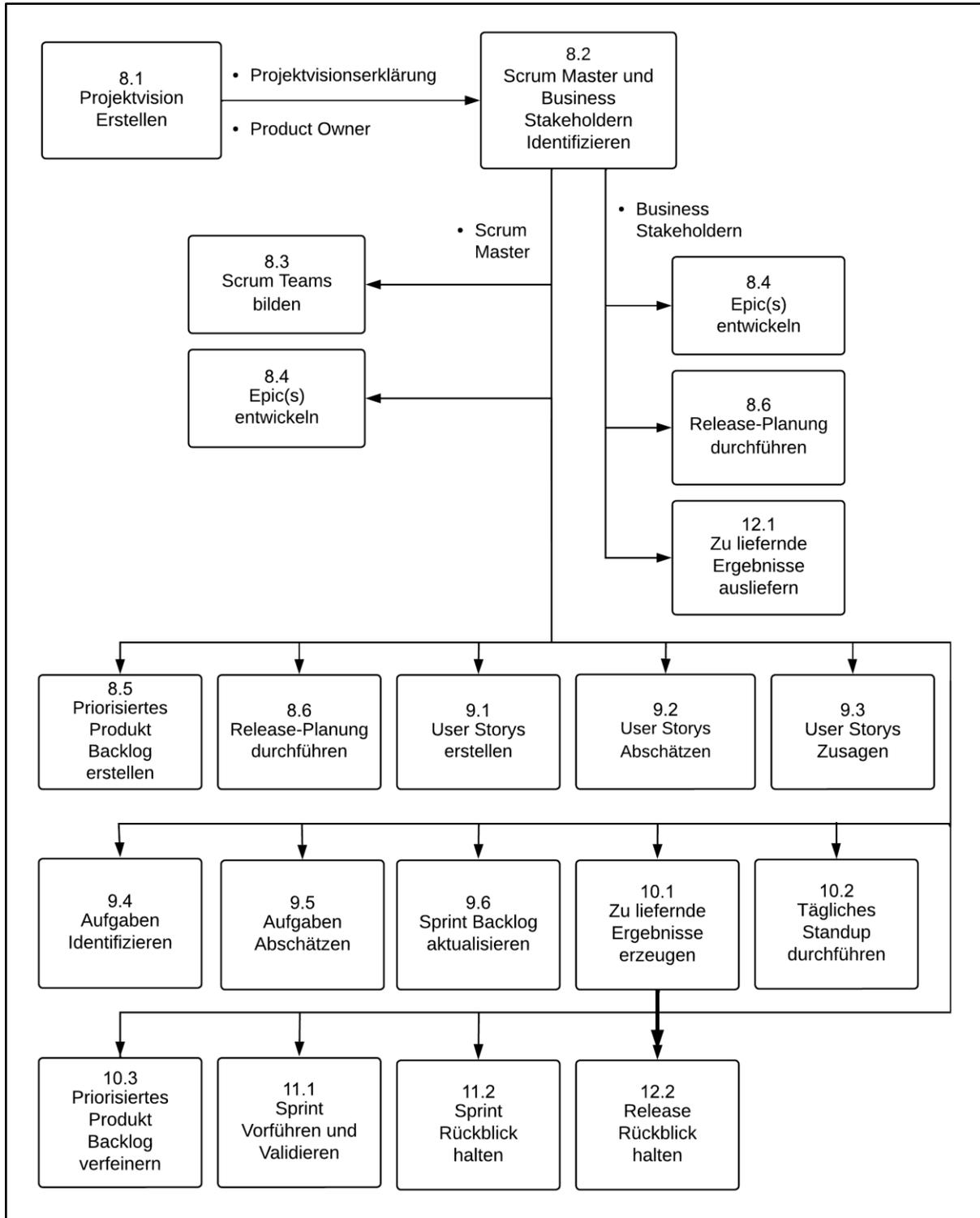


Abbildung 8-7: Scrum Master und Business Stakeholdern Identifizieren—Datenflussdiagramm

8.2.1 Inputs

8.2.1.1 Product Owner*

Siehe Abschnitt 8.1.3.1.

8.2.1.2 Projektvisionserklärung*

Siehe Abschnitt 8.1.3.2.

8.2.1.3 Personalanforderungen

Die Festlegung der erforderlichen Fähigkeiten des Scrum Masters ist einer der ersten Schritte, die durchgeführt werden müssen, bevor ein Scrum Team vollständig gebildet werden kann (siehe Prozess 8.3). Es ist wichtig, die Rolle und die Verantwortlichkeiten der Scrum Master Position sowie die gewünschten Kenntnisse und Fähigkeiten zu dokumentieren, die für den Erfolg eines Projekts erforderlich sind. Normalerweise arbeitet der Product Owner mit der Personalabteilung oder anderen relevanten internen Business Stakeholdern zusammen, um die Rollenbeschreibung des Scrum Masters festzulegen und abzuschließen.

Außerdem werden die wichtigsten Business Stakeholdern identifiziert und ihre Rolle im Projekt geklärt.

8.2.1.4 Verfügbarkeit und Zusage des Personals

Vor der Auswahl des Scrum Masters und der Business Stakeholdern muss deren Verfügbarkeit verifiziert werden. Nur Teammitglieder, welche verfügbar sind und sich dem Projekt voll verschreiben können sollten ausgewählt werden. Die Verfügbarkeit und Zusage des Personals wird üblicherweise in Form eines Kalenders dokumentiert, der anzeigt, wann menschliche Ressourcen für die gesamte Aktivphase des Projekts verfügbar sind.

Um effektiv agieren zu können, sollten Scrum Teams idealerweise sechs bis zehn Mitglieder haben; das Ersetzen von Personen oder Austauschen von Teammitgliedern ist in Scrum Kernteam nicht ratsam. Deshalb ist es wichtig, im Scrum Kernteam Personen zu haben, welche verfügbar sind und sich dem Projekt voll verschreiben können.

8.2.1.5 Organisatorische Ressourcen-Matrix

Die Organisatorische Ressourcen-Matrix ist eine hierarchische Wiedergabe einer Kombination aus funktionaler Organisationsstruktur und projektspezifischer Organisationsstruktur. Matrix-Organisationen bringen Teammitglieder aus unterschiedlichen funktionalen Abteilungen in das Projekt ein, wie beispielsweise aus der Informationstechnologie sowie der Finanz-, Marketing, Verkauf-, Produktions- und anderer Abteilungen – und erzeugen damit funktionsübergreifende Teams.

Teammitglieder in einer Matrix-Organisation erfüllen zwei Ziele – Funktionalität und Projektbezug. Teammitglieder werden bei projektbezogenen Aktivitäten durch den Product Owner angeleitet, während die funktionellen Manager ihre Arbeit im Bezug auf ihre Abteilungen durchführen, in dem sie beispielsweise Leistungsbewertungen durchführen und Urlaub genehmigen.

8.2.1.6 Fähigkeiten-Anforderungs-Matrix

Die Fähigkeiten-Anforderungs-Matrix, auch als Kompetenzrahmen bekannt, wird verwendet, um Fähigkeitsdefizite und die Anforderungen für die Weiterbildung der Teammitglieder einzuschätzen. Eine Fähigkeiten-Matrix verzeichnet die Fähigkeiten, Möglichkeiten und das Interesse der Teammitglieder an der Nutzung dieser Fähigkeiten und Möglichkeiten für ein Projekt. Anhand der Matrix kann die Organisation jedes Fähigkeitsdefizit eines Teammitglieds einschätzen und die Angestellten identifizieren, welche Weiterbildung in spezifischen Bereichen oder Kompetenzen benötigen.

8.2.1.7 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11.

8.2.2 Hilfsmittel

8.2.2.1 Auswahlkriterien*

Die Auswahl von passenden Scrum Master(n) und die Identifizierung relevanter Business Stakeholdern ist von essentieller Bedeutung für den Erfolg eines jeden Projekts. In einigen Projekten kann es hierbei Voraussetzungen geben, welche bestimmte Teammitglieder auf ihre Rollen festlegt.

Wenn es die Möglichkeit der Auswahl eines Scrum Masters gibt, lauten die wichtigen Auswahlkriterien wie folgt:

1. *Fähigkeiten zur Problemlösung*–Dies ist eine der wichtigsten Kriterien bei der Auswahl des Scrum Masters. Der Scrum Master sollte die nötigen Fähigkeiten und die nötige Erfahrung haben, um jegliche Impedimente für das Team zu beseitigen.

2. *Verfügbarkeit*—Der Scrum Master sollte verfügbar sein, um die verschiedenen Meetings, darunter das Release-Planungs Meeting, das tägliche Standup Meeting und andere sprintbezogene Meetings abzuhalten
3. *Einsatzwille*—Der Scrum Master sollte mit vollem Einsatz dafür sorgen, dass das Scrum Team mit einem förderlichen Arbeitsumfeld versehen ist, um die erfolgreiche Durchführung des Scrum-Projekts zu gewährleisten.
4. *Unterstützende Führung-Stil*—Abschnitt 3.10.4.1 bietet hierzu weitere Details

Beim Identifizieren der Business Stakeholdern ist es wichtig zu beachten, dass Business Stakeholdern die Gesamtheit der Kunden, Anwender und Sponsoren sind, welche sich regelmäßig mit dem Product Owner, dem Scrum Master und dem Scrum Team in Verbindung setzt, um Vorgaben zu liefern und die Erzeugung des Produkts des Projekts zu leiten. Die Business Stakeholdern beeinflussen das Projekt während seiner gesamten Aktivphase.

8.2.2.2 Expertise der Personalabteilung

Expertise von den Personalmanagern (Human Resource Managers) kann bei der Identifikation des Scrum Masters und der Business Stakeholdern wertvoll sein. Die Personalabteilung besitzt spezialisiertes Wissen über die Arbeitnehmer einer Organisation und verschiedene Methoden, welche mitunter dabei helfen können, den Scrum Master und die Business Stakeholdern zu identifizieren.

8.2.2.3 Ausbildung

Scrum unterscheidet sich radikal von den traditionellen Methoden des Projektmanagements. Es kann sein, dass Teammitglieder nicht immer das benötigte Wissen oder die Fähigkeiten besitzen, um in einer Scrum-Umgebung zu arbeiten. Der Product Owner sollte die Notwendigkeit von Weiterbildung der potentiellen Teammitglieder einschätzen und diese Weiterbildung ermöglichen, um Wissenslücken im Team zu beseitigen. Der Product Owner ist normalerweise für die Bewertung und Auswahl der Teammitglieder verantwortlich, aber spricht dies häufig mit dem Scrum Master ab, welcher zusätzliches Wissen über die Ressourcen dank seiner Arbeit an anderen Projekten besitzen kann.

Angemessenes Training sollte für die Mitgliedern des Scrum Teams sowohl vor Beginn der Arbeit als auch während der Arbeit an den Projekten bereitgestellt werden. Mitglieder des Scrum Teams sollten bereit sein, voneinander und von erfahreneren Personen im Team zu lernen.

8.2.2.4 Ressourcenkosten

Eine der primären Überlegungen bei Auswahl des Personals steht in Verbindung mit der Abwägung zwischen Erfahrung und Gehaltskosten. Es gibt andere personenbezogene Faktoren, die die Kosten beeinflussen, welche mitunter ebenfalls berücksichtigt werden müssen. Idealerweise sollten Scrum Master, Teammitglieder und Business Stakeholdern nahe beieinander arbeiten, sodass sie regelmässig und leicht miteinander kommunizieren können. Wenn dieses nicht möglich ist, und es verteilte Teams gibt, müssen zusätzliche Ressourcen zur Unterstützung von Kommunikationswegen, dem Verständnis von kulturellen Verschiedenheiten, der Synchronisation von Arbeit und der Verteilung von Wissen bereit gestellt werden.

8.2.3 Outputs

8.2.3.1 Identifizierter Scrum Master*

8

Der Scrum Master ist ein Moderator und "dienender Leader", der sicherstellt, dass das Scrum Team zur erfolgreichen Fertigstellung des Projekts mit einer Umgebung versorgt wird, die dem erfolgreichen Abschluss des Projekts förderlich ist. Der Scrum Master leitet an, moderiert und lehrt die Scrum-Praktiken allen am Projekt Beteiligten; beseitigt Impedimente für das Team; und stellt sicher dass die Scrum Prozesse richtig angewendet werden. Es ist die Aufgabe des Product Owners, den Scrum Master für ein Scrum Projekt auszuwählen.

Die Rolle des Scrum Masters ist detaillierter in Abschnitt 3.4 beschrieben.

8.2.3.2 Identifizierte(r) Business Stakeholdern*

Business Stakeholdern, ein Sammelbegriff, der Kunden, Anwender und Sponsoren umfasst, setzen sich regelmäßig mit dem Scrum Kernteam in Verbindung und beeinflussen das Projekt während des gesamten Prozesses der Produktentwicklung. Die Business Stakeholdern sind diejenigen, für die das Projekt gemeinschaftliche Vorteile generiert.

Die Rolle des Business Stakeholders wird in Abschnitt 3.3.2 beschrieben.

8.3 Scrum Team bilden

In diesem Prozess werden die Mitglieder des Scrum-Teams auf der Grundlage der Fähigkeiten ermittelt, die für die Fertigstellung der Projektleistungen erforderlich sind, sowie unter Berücksichtigung der Verfügbarkeit, der Kosten und der Soft Skills, die für die Mitglieder eines Scrum-Teams wichtig sind. Üblicherweise ist der Product Owner für die Auswahl der Teammitglieder verantwortlich, doch oft geschieht dies in Zusammenarbeit mit dem Scrum Master.

Abbildung 8-8 zeigt Inputs, Hilfsmittel und Outputs für den *Scrum Team Bilden* Prozess.

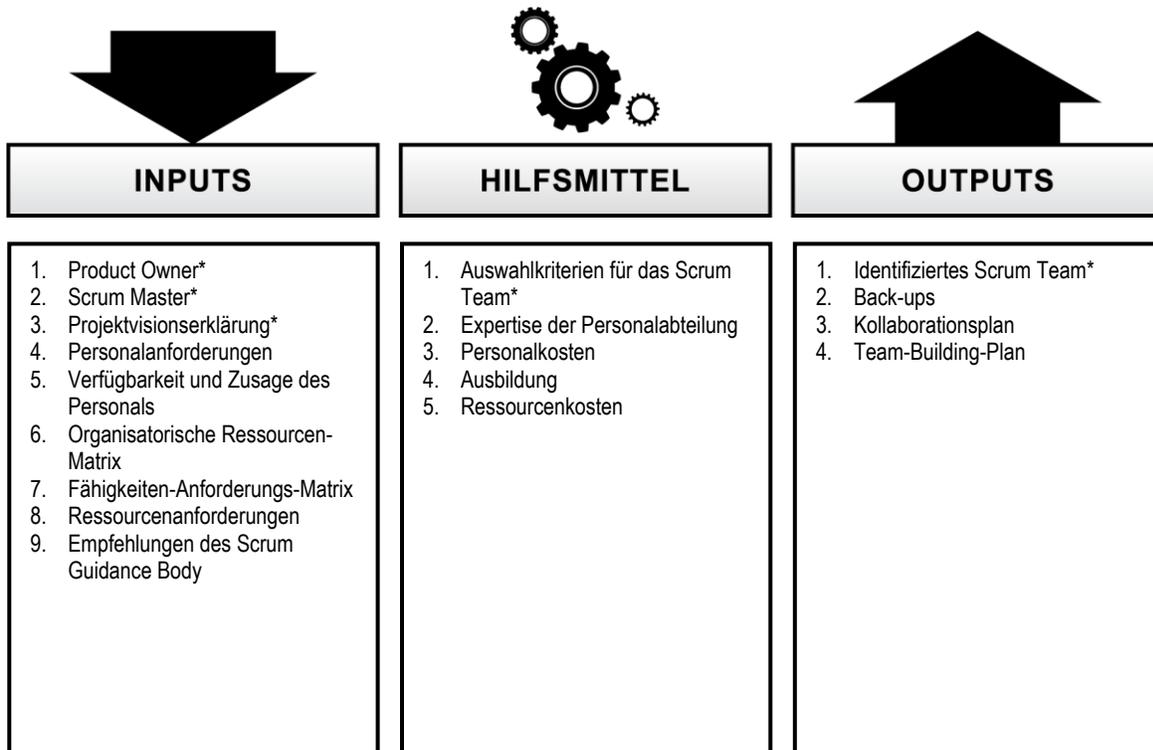


Abbildung 8-8: Scrum Team bilden–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen (*) bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess

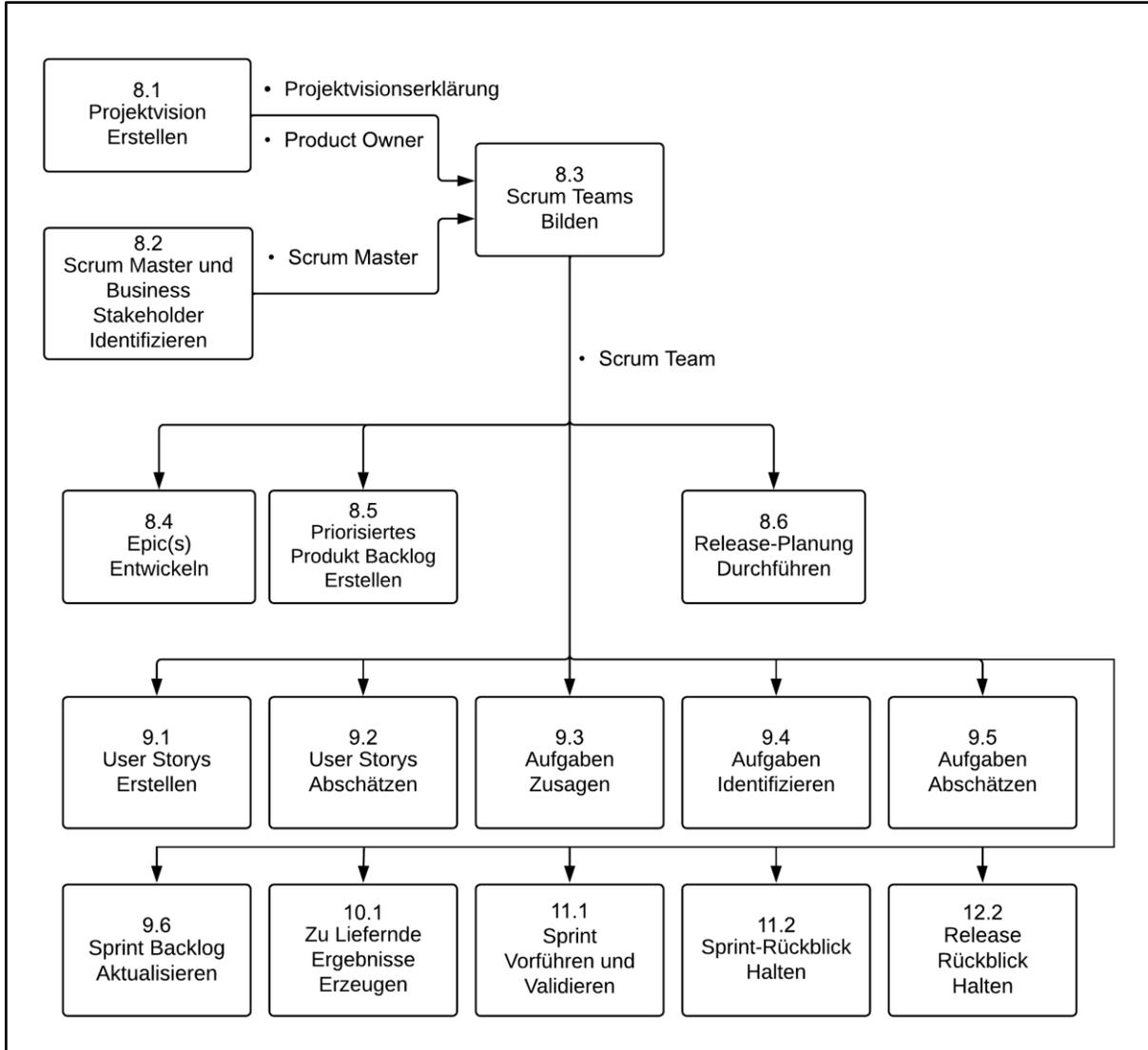


Abbildung 8-9: Scrum Team bilden-Datenflussdiagramm

8.3.1 Inputs

8.3.1.1 Product Owner*

Siehe Abschnitt 8.1.3.1.

8.3.1.2 Scrum Master*

Siehe Abschnitt 8.2.3.1.

8.3.1.3 Projektvisionserklärung*

Siehe Abschnitt 8.1.3.2.

8.3.1.4 Personalanforderungen

Siehe Abschnitt 8.2.1.8.

8.3.1.5 Verfügbarkeit und Zusage des Personals

Siehe Abschnitt 8.2.1.9.

8.3.1.6 Organisatorische Ressourcen-Matrix

Siehe Abschnitt 8.2.1.10.

8.3.1.7 Fähigkeiten-Anforderungs-Matrix

Siehe Abschnitt 8.2.1.11.

8.3.1.8 Anforderungen an Ressourcen

Diese Anforderungen beinhalten sämtliche Ressourcen (mit Ausnahme von Menschen), welche vom Scrum Team benötigt werden, um effektiv zu funktionieren. Diese Ressourcen beinhalten Büro-Infrastruktur, Treffpunkte, Arbeitsmaterial, das Scrumboard etc. Im Falle von virtuellen Teams müssen weitere Ressourcen wie Kollaborations Hilfsmittel, Videokonferenz-Systeme, Dokumentenmanagementsystemen, Übersetzungsdienstleistungen etc. in Betracht gezogen werden.

8.3.1.9 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11.

8.3.2 Hilfsmittel

8.3.2.1 Auswahlkriterien für das Scrum Team*

Das Scrum Team ist der Kern eines jeden Scrum Projekts und das Zusammenziehen der richtigen Teammitglieder ist wichtig für die erfolgreiche Fertigstellung des Scrum Projekts. Mitglieder des Scrum Team sind Allrounder-Spezialisten in dem Sinne, dass sie Wissen aus einer Vielzahl von Bereichen haben und in mindestens einem Bereich Experten sind. Über ihre Expertise zur Materie hinaus sind es die Sozialkompetenzen der Teammitglieder, die den Erfolg eines Selbst-organisierenden Teams ausmachen.

Ideale Mitglieder eines Scrum Teams sind unabhängig, eigenmotiviert, kundenorientiert, verantwortungsbewusst, und gemeinschaftsorientiert. Das Team sollte in der Lage sein, ein Umfeld des unabhängigen Denkens und der gemeinschaftlichen Entscheidungsfindung zu fördern und aus dieser Struktur ein Maximum an Vorteilen zu ziehen.

8.3.2.2 Expertise der Personalabteilung

Expertise von den Personalmanagern (Human Resource Managers) kann bei der Formierung eines Scrum Teams wertvoll sein. Die Personalabteilung besitzt spezialisiertes Wissen über die Arbeitnehmer einer Organisation und verschiedene Methoden, welche mitunter dem Product Owner, Scrum Master und den Sponsoren dabei helfen können, die richtigen Teammitglieder zu identifizieren.

8.3.2.3 Personalkosten

Alle Kosten, welche mit den Personalanforderungen verbunden sind, müssen bewertet, analysiert, genehmigt, und ins Budget aufgenommen werden.

8.3.2.4 Ausbildung

Es kann sein, dass Teammitglieder nicht immer das benötigte Wissen oder die Fähigkeiten besitzen, um spezielle Aufgaben auszuführen. Der Product Owner und/oder Scrum Master sollte den Schulungsbedarf potenzieller Teammitglieder ermitteln. Sobald die Kandidaten ausgewählt sind, sollte für alle festgestellten Fähigkeits- oder Wissenslücken die entsprechende Schulung angeboten werden.

Für eine tatsächlich effektive Umsetzung von Scrum ist es notwendig, dass es ein klares Bewusstsein innerhalb der Organisation für die Prinzipien und Werte von Scrum gibt. Dieses Bewusstsein wird bei der erfolgreichen Umsetzung von Scrum helfen. Das Scrum Team muss für die Praktiken von Scrum sensibilisiert und ausgebildet werden, und der Scrum Master sollte dabei die Rolle des Trainers für das Team übernehmen. Da die Planung eines Sprints ein wesentlicher Erfolgsfaktor ist, wird die Ausbildung den Teams zu verstehen helfen, wie erreichbare Ziele für einen Sprint zu besprechen und zu identifizieren sind.

Der Scrum Master muss das Beste aus dem Scrum Team holen, indem er es während des Entwicklungsprozesses motiviert und anleitet. Mittels des Trainings und der Ausbildung der Teammitglieder kann der Scrum Master bei der Klarstellung von Angelegenheiten und Herausforderungen helfen, die diesen begegnen. Normalerweise werden jegliche Angelegenheiten und Konflikte, welche innerhalb des Teams auftreten, von diesem mit der Unterstützung und dem Training des Scrum Masters, wenn dies benötigt wird, gelöst. Der Scrum Master sollte Angelegenheiten wie niedrige Moral oder Mangel an Koordination innerhalb des Teams adressieren. Sie/er ist verantwortlich für das Beseitigen von Impedimenten für das Team. Wenn nötig, wird der Scrum Master externe Probleme und Impedimente an das Management zur Lösung und Beseitigung weitergeben.

Weiterbildung und Weiterbildungskosten werden auch beim Prozess zur Identifizierung von Scrum Master und Business Stakeholdern identifizieren in Abschnitt 8.2.2.3 behandelt.

8.3.2.5 Ressourcenkosten

Die Kosten, die in Zusammenhang mit non-personellen Anforderungen entstehen, müssen eingeschätzt, analysiert, bewilligt und ins Budget aufgenommen werden. Eine Resource ist im Rahmen eines Projekts alles, was zur Durchführung der Aufgabe oder Aktivität genutzt wird; dies umfasst unter anderem (aber nicht nur) Ausrüstung, Material, Arbeit, externe Dienstleistungen und Räumlichkeiten.

8.3.2.6 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

8.3.3 Outputs

8.3.3.1 Identifiziertes Scrum Team*

Das Scrum Team, mitunter auch als Entwicklungsteam bezeichnet, ist eine Gruppe oder ein Team von Personen, welche dafür verantwortlich sind, die geschäftlichen Anforderungen nach der Spezifikation des Product Owners zu verstehen, die User Storys abzuschätzen, und schließlich die zu liefernden Ergebnisse des Projekts zu erzeugen. Scrum Teams sind funktionsübergreifend und selbst-organisierend. Das Team entscheidet über die Menge an Arbeit, die für einen Sprint zugesagt wird und ermittelt den besten Weg, diese Arbeit zu leisten. Das Scrum Team besteht aus funktionsübergreifenden Teammitgliedern, welche die gesamte Arbeit für die Erzeugung von potenziell versandfähigen lieferbaren Ergebnissen übernehmen; dies umfasst Entwicklung, Test, Qualitätssicherung etc.

Die Identifizierung des Scrum Teams ist die Aufgabe des Product Owners, der sich hierbei oft mit dem Scrum Master abspricht.

Die Rolle des Scrum Teams wird detaillierter in Abschnitt 3.6 beschrieben.

8.3.3.2 Back-ups

Bei der Auswahl von Teammitgliedern ist es wichtig, Backups für kritische Fähigkeiten zu identifizieren, vorzugsweise innerhalb desselben Scrum-Teams. Obwohl die Verfügbarkeit und die Zusage eines jeden Teammitglieds vorher bestätigt wird, können Ereignisse wie beispielsweise Krankheit, Familiäre Notfälle oder das Ausscheiden des Teammitglieds aus der Organisation, auftreten. Scrum Teams arbeiten in kleinen Gruppen von sechs bis zehn Personen, und das Bestehen von Backups sichert ab, dass es keinen signifikanten Verlust an Produktivität bei Verlust eines Teammitglieds gibt.

8.3.3.3 Kollaborationsplan

Kollaboration ist ein äußerst bedeutsames Element von Scrum und die Planung der Interaktion und Kollaboration zwischen Entscheidungsträgern, Business Stakeholdern und Teammitgliedern ist unverzichtbar. Der Kollaborationsplan ist ein optionaler Output und kann sowohl formell als auch informell sein. Mitunter besteht er einfach nur aus einer mündlichen Verständigung zwischen den verschiedenen Business Stakeholdern, da Scrum unnötige Dokumentation vermeidet. Allerdings kann es sein, dass für größere, komplexere Projekte, insbesondere bei verteilten Teams, eine formelere Vereinbarung notwendig wird.

Der Plan kann die Art und Weise behandeln, mit der die Scrum Kernteam-Mitglieder, Business Stakeholdern und andere in das Scrum Projekt involvierte Personen im gesamten Verlauf des Projekts miteinander kommunizieren und kollaborieren, und kann darüber hinaus spezifische Hilfsmittel und Methoden definieren, welche zu diesem Zweck angewendet werden.

In verteilten Teams kann es beispielsweise nötig werden, eine Vereinbarung zu treffen, wann und wie Meetings abgehalten werden, welche Form von Kommunikationsmitteln genutzt werden und wer bei jedem spezifischen Meeting beteiligt sein sollte.

8.3.3.4 Team-Building Plan

Da das Scrum Team funktionsübergreifend ist muss jedes Mitglied aktiv an allen Aspekten des Projekts teilnehmen. Der Scrum Master sollte Probleme gemeinsam mit den Teammitgliedern identifizieren und diese sorgfältig bearbeiten, um ein effektives Team zu erhalten.

Um den Teamzusammenhalt zu stärken, sollte der Scrum Master sicherstellen, dass das Verhältnis zwischen den Teammitgliedern positiv ist und dass die Teammitglieder zum Erreichen der Endziele von Projekt und Organisation zusammenhalten, um so zu einer größeren Effizienz und verstärkter Produktivität zu führen.

In diesem Zusammenhang ist es wichtig, Abschnitt 3.10 zu beachten, der populäre Personalmanagement-Theorien und deren Relevanz für Scrum behandelt.

8.4 Epic(s) entwickeln

In diesem Prozess dient die Projektvisionserklärung als Grundlage für die Entwicklung von Epics, die die High-Level-Anforderungen für das Projekt definieren. Der Product Owner kann User Group Meetings und andere Tools nutzen, um die Anforderungen der Business Stakeholdern zu sammeln.

Abbildung 8-10 zeigt um so Inputs, Hilfsmittel und Outputs für *Epic(s) entwickeln* Prozess.

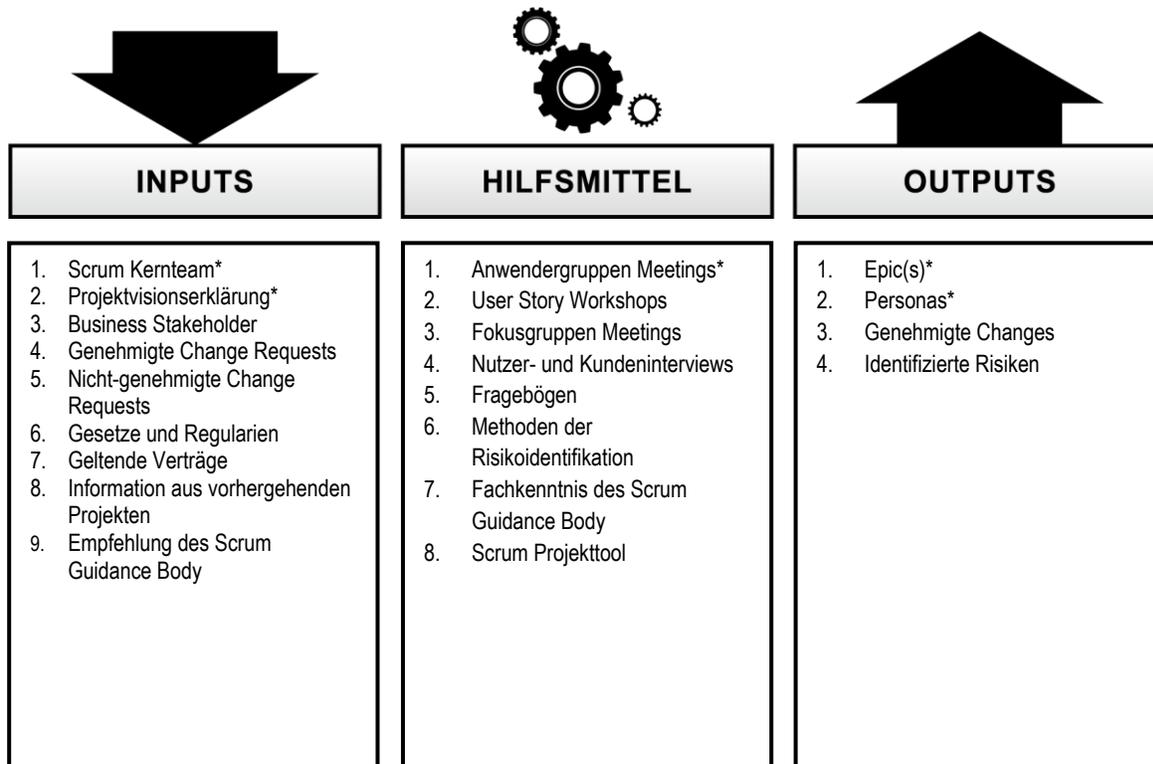


Abbildung 8-10: Epic(s) entwickeln–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

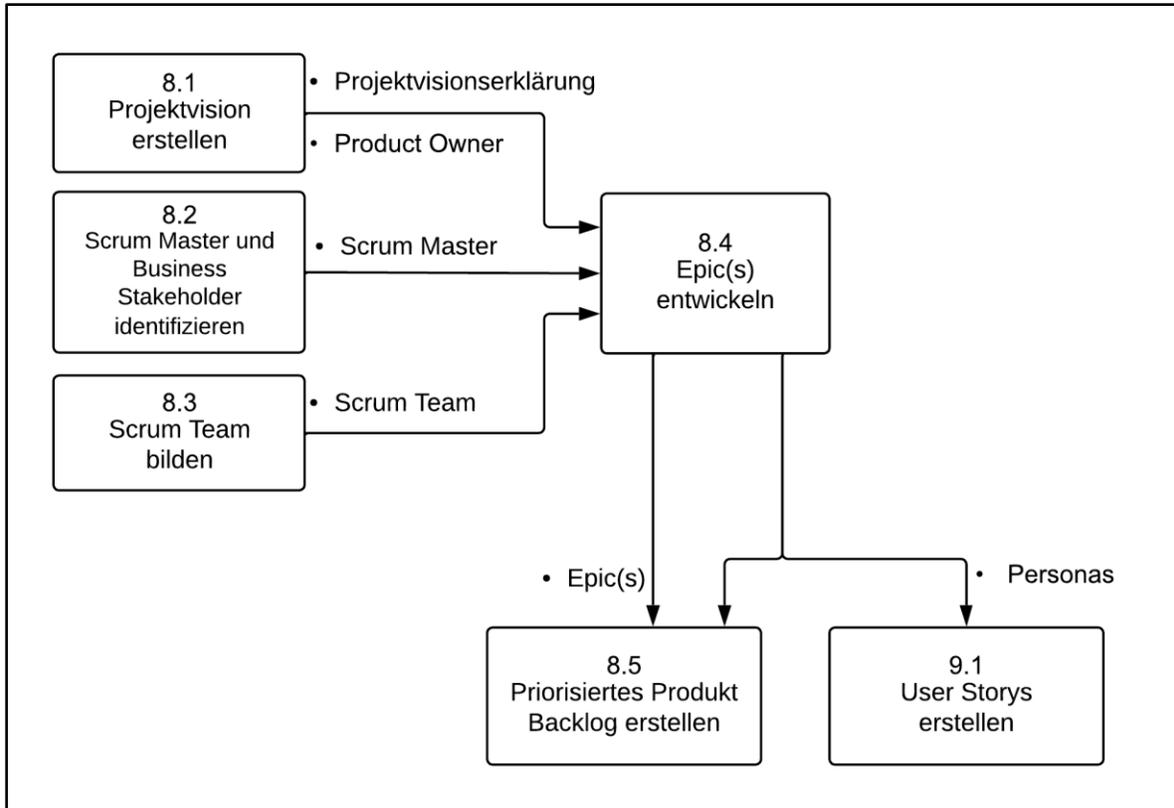


Abbildung 8-11: Epic(s) entwickeln-Datenflussdiagramm

8.4.1 Inputs

8.4.1.1 Scrum Kernteam*

Das Scrum Kernteam besteht aus dem Scrum Team, dem Scrum Master und dem Product Owner, vergleiche hierzu Abschnitt 3.3.1.

8.4.1.2 Projektvisionserklärung*

Siehe Abschnitt 8.1.3.2.

8.4.1.3 Business Stakeholdern

Siehe Abschnitt 8.2.3.2.

8.4.1.4 Genehmigte Change Requests

Genehmigte Change Requests, die dem Programm oder Portfolio entstammen, stellen Vorgaben dar, welche der Liste von genehmigten Projektänderungen, welche in zukünftigen Sprints implementiert werden sollen, hinzugefügt werden. Jede Änderung kann ihr eigenes Epic oder ihre eigene User Story benötigen und kann damit ein Input für den Prozess *Epic(s) entwickeln* werden. Genehmigte Change Requests zu diesem Prozess können auch von anderen Scrum Prozessen stammen.

Change Requests und genehmigte Change Requests werden in den Abschnitten 6.3.1, 6.4.2.1 und 6.6 behandelt.

8.4.1.5 Nicht genehmigte Change Requests

Anfragen für Änderungen werden üblicherweise als Change Requests eingereicht und bleiben nicht genehmigt, bis die Zustimmung formell geschieht. Genehmigte Change Requests und Nicht genehmigte Change Requests für den Prozess *Epic(s) entwickeln* können aus *Zu liefernden Ergebnisse erzeugen*, *Tägliches Standup durchführen* und anderen Prozessen stammen.

Change Requests und genehmigte Change Requests werden in den Abschnitten 6.3.1, 6.4.2.1 und 6.6 behandelt.

8.4.1.6 Gesetze und Regulierungen

Je nach Projekt kann es von kontrollierenden Behörden festgelegte Gesetze und Verordnungen geben, welche die Planung und Ausführung des Projekts beeinflussen. Gesetze sind extern und von einer staatlichen Institution festgelegt. Verordnungen können sowohl extern als auch intern sein. Interne Verordnungen sind solche, welche innerhalb des Unternehmens, typischerweise auf Basis von Firmenrichtlinien, zur Anwendung kommen. Diese Verordnungen können sich auf Qualitätsmanagementsysteme, Finanz- oder Personalrichtlinien etc. beziehen. Externe Verordnungen sind solche, welche sich auf gesetzlich festgelegte Standards, Normen oder Anforderungen beziehen.

Gesetze und Verordnungen müssen bei der Entwicklung von Epics beachtet werden. Epics basieren auf geschäftlichen Anforderungen, und um diese Anforderungen zu erfüllen muss das Projektteam sich sowohl an interne als auch externe Gesetze und Verordnungen halten.

Mitunter können einige Gesetze und Verordnungen, welche mehrere Scrum Projekte beeinflussen, in die Empfehlungen des Scrum Guidance Body aufgenommen werden. Vergleiche hierzu Abschnitt 8.1.1.11.

8.4.1.7 Geltende Verträge

Wenn das Projekt oder einige seiner Teile im Rahmen eines Vertrags durchgeführt werden, definiert dieser Vertrag den Umfang der Arbeit und die Art des Vertrages, und beeinflusst das Risiko des Projekts. Es ist nicht unbedingt das gesamte Projekt, das im Rahmen eines Vertrags durchgeführt wird.

Es existiert eine Vielzahl von Verträgen. Einige der verbreitetsten Verträgen bei Scrum Projekten werden im Folgenden aufgeführt:

Vertrag zur inkrementellen Bereitstellung – Dieser Vertrag beinhaltet Inspektionen in regelmäßigen Abständen. Dies hilft dem Kunden oder den Business Stakeholdern beim Treffen von Entscheidungen im Bezug auf Produktentwicklung, welche periodisch an jedem Inspektionenpunkt erfolgen. Der Kunde kann die Entwicklung des Produkts entweder akzeptieren, sich entscheiden, diese zu stoppen, oder Änderungen am Produkt fordern.

Joint Venture Vertrag — Dieser Vertrag wird üblicherweise genutzt, wenn zwei oder mehr Partnerparteien sich an der Arbeit des Projekts beteiligen. Die am Projekt beteiligten Parteien erreichen beide ein gewisses Maß an Investitionsrendite, da die erzielten Gewinne oder Vorteile zwischen den Parteien geteilt werden.

Vertrag zur Entwicklung in Phasen — Dieser Vertrag ermöglicht die monatliche/ vierteljährliche Verfügbarmachung der Finanzierung, nachdem ein Release erfolgreich wurde. Er gibt dabei Anreize sowohl für Kunden als auch Lieferanten und sichert ab, dass das finanzielle Risiko des Kunden auf diesen spezifischen Zeitraum begrenzt ist, da nicht erfolgreiche Releases nicht finanziert werden.

Vertrag mit Anreizen und Geldbußen — Diese Verträge basieren auf einer Vereinbarung, dass der Lieferant eine finanzielle Belohnung erhält, wenn das Produkt des Projekts innerhalb des Zeitrahmens geliefert wird, aber Geldbußen hinnehmen muss, sollte die Lieferung zu spät erfolgen.

Andere beliebte Vertragsvarianten umfassen Verträge für das Bezahlen nach Eigenschaften, Zeit- und Materialverträge, sowie Verträge mit einem Festpreis, einem festen Lieferumfang oder festgelegtem Profit.

Epic(s) sollten unter Beachtung der Bedingungen des jeweilig genutzten Vertragstyps entwickelt werden.

8.4.1.8 Informationen von vorhergehenden Projekten

Informationen und Einsichten, welche auf ähnlichen, vorhergehenden Projekten innerhalb der Organisation erwachsen sind, stellen wertvolle Vorgaben für das Entwickeln von Epic(s) und das Bewerten von Risiken dar. Informationen aus vorherigen Projekten können die Notizen des Projektmanagers, die Projektlogs, und Bemerkungen der Business Stakeholder beinhalten.

Einige der Informationen und besten Methoden im Bezug auf Information aus vorherigen Projekten können auch über die Empfehlungen des Scrum Guidance Body verfügbar sein.

8.4.1.9 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11

Bei der Entwicklung von Epic(s) können die Empfehlungen des Scrum Guidance Body Informationen über Regeln, Verordnungen, Standards und beste Methoden zur und bei der Entwicklung von Epic(s) beinhalten.

8.4.2 Hilfsmittel

8.4.2.1 Anwender-Gruppen Meetings*

Anwender-Gruppen Meetings versorgen den Product Owner mit Informationen aus erster Hand über die Erwartungen relevanter Business Stakeholdern (vorwiegend Anwender oder Kunden des Produkts). An den User Group Meetings können auch relevante Mitglieder des Scrum Core Teams teilnehmen. Dies hilft bei der Formulierung von Abnahmekriterien für das Produkt, und ermöglicht wertvolle Einsichten für die Entwicklung von Epic(s). Diese Meetings fördern auch die Akzeptanz des Projekts, und erzeugen ein gemeinsames Verständnis zwischen Scrum Kernteam und relevanten Business Stakeholdern(n). Anwender-Gruppen Meetings sind für die Prävention kostenaufwändiger Nacharbeiten, welche aus Unklarheiten in Bezug auf Erwartungen und Anforderung entstehen können, von großer Bedeutung.

8.4.2.2 User Story Workshops

User Story Workshops werden während der Entwicklung von Epics abgehalten. Der Scrum Master leitet diese Sitzungen an, an denen das gesamte Scrum Kernteam und mitunter, wenn dies wünschenswert ist, andere Business Stakeholdern teilnehmen. Diese Workshops helfen dem Product Owner bei der Priorisierung von Anforderungen und ermöglicht es dem Scrum Kernteam ein gemeinsames Verständnis für die Abnahmekriterien zu entwickeln. Dies sichert ab, dass die Epic(s) und die User Storys die Funktionalität aus Sicht des Anwenders beschreiben, einfach verständlich sind, und zuverlässig abgeschätzt werden können. User Story Workshops sind nützlich für das Verstehen von Anwendererwartungen an das zu liefernde Ergebnis, und sind hervorragend für das Team-Building. Darüber hinaus erleichtern sie auch die Planung des nächsten Sprints. Ein User Story Workshop ist eine gute Plattform, um jedes Element eines Produktes zu besprechen und zu klären, wobei oft die kleinsten Kleinigkeiten zur Sicherstellung von Klarheit besprochen werden.

8.4.2.3 Fokusgruppen Meetings

Fokusgruppen versammeln Individuen in einer geleiteten Sitzung, um Ihre Meinungen, Wahrnehmungen oder Bewertungen eines Produkts, einer Dienstleistung, oder eines erwünschten Ergebnisses zu verlautbaren. Mitglieder von Fokusgruppen haben die Freiheit, sich gegenseitig Fragen zu stellen und Unklarheiten in spezifischen Bereichen oder Konzepten zu beseitigen.

Durch Fragen, konstruktive Kritik und Feedback führen Fokusgruppen zu einer besseren Produktqualität und tragen damit zum Erfüllen der Anwendererwartungen bei. In diesen Meetings finden die Mitglieder der Fokusgruppen mitunter einen Konsens für bestimmte Bereiche, während bei anderen die Meinungen auseinandergehen können. Wenn Mitglieder unterschiedliche Meinungen oder Perspektiven haben, muss unbedingt versucht werden, diese Unterschiede zu überbrücken und zu einem Konsens zu gelangen.

Fokusgruppen Sessions können den Teams dabei helfen, Innovative Ideen zu entwickeln, Probleme zu lösen und Vorschläge für Verbesserungen zu machen. Diese Meetings ermöglichen die Tatsachenfeststellung und generieren Ideen und Feedback von potentiellen Anwendern und Produktentwicklern. Diese Meetings werden üblicherweise zur Planung, Bewertung und Verbesserung eines Produkts oder einer Dienstleistung durchgeführt. Einsichten, welche durch diese Meetings erreicht werden, können darüber hinaus auch bei der Entwicklung von User Storys und Epic(s) helfen. Mitunter werden Fokusgruppen Meetings durchgeführt, um Probleme zu lösen, welche bei der Entwicklung von Epic(s) entstanden sind.

8.4.2.4 Anwender- und Kundeninterviews

Die Beteiligung von Business Stakeholdern, darunter die Sponsoren, die Anwender und die Kunden des Produkts, ist wichtig, um den nötigen Kontext und die nötigen Einsichten für die Entwicklung von Epics zu erhalten. Sinnvoll aufgewendete Zeit, in der die Anwender und Kunden befragt werden, resultiert in der Sicherheit, dass die Anforderungen der Epics kongruent zur Gesamtvision des Projekts sind und damit einen höheren Wert einbringen.

Diese Interviews helfen bei:

- Der Identifizierung und dem Verstehen der Bedürfnisse/Erwartungen der Business Stakeholdern
- Der Sammlung von Meinungen und Fakten
- Dem Verständnis für die Perspektive der Business Stakeholdern hinsichtlich des Endprodukts
- Der Sammlung von Feedback über ein wiederholtes/teilweise entwickeltes Produkt.
- Der Einbindung und dem Engagement von Anwendern und/oder Kunden.

8.4.2.5 Fragebögen

Ein kosteneffektiver Weg zur Sammlung von quantitativen und qualitativen statistischen Einsichten von einer großen Anzahl an Anwendern oder Kunden ist die Verwendung von Fragebögen. Ein Fragebogen ist ein Forschungsinstrument, bei dem die Befragten vorbereitete Fragen beantworten, damit Informationen über ein spezifisches Problem oder Thema gesammelt werden können. Fragebögen können sowohl durch den Befragten selbst als auch mittels eines Interviewers ausgefüllt werden.

Bei der Entwicklung von Fragebögen muss äußerste Sorgfalt gewahrt werden, um die richtige Zielgruppe auszuwählen und die richtige Methode für die Durchführung der Umfrage zu finden, um Fehler und Voreingenommenheiten zu vermeiden.

Bei der Entwicklung von Epics können der Product Owner oder der Scrum Master eine Umfrage durchführen, um relevante Informationen von den Business Stakeholdern oder dem Scrum Team zu erhalten.

8.4.2.6 Methoden der Risikoidentifikation

Siehe Abschnitt 7.4.1.1

8.4.2.7 Fachkenntnis des Scrum Guidance Body

Siehe Abschnitt 3.3.2

Der Scrum Guidance Body (SGB) ist eine ist eine optionale Rolle. Er besteht üblicherweise aus einer Gruppe von Dokumenten und/oder einer Gruppe von Experten, welche typischerweise an der Definition von Zielen im Bezug auf Qualität, staatliche Vorgaben, Sicherheit und andere organisatorische Schlüsselparameter beteiligt sind. Diese Ziele stellen Richtlinien für die Arbeit des Product Owners, des Scrum Masters und des Scrum Teams bereit. Der Scrum Guidance Body kann auch dabei helfen, die besten Maßnahmen festzuhalten, welche bei allen Scrum Projekten der Organisation genutzt werden sollten.

Der Scrum Guidance Body trifft keine Entscheidungen im Bezug auf das Projekt. Stattdessen fungiert er als ratgebende oder anleitende Struktur für alle Hierarchie-Ebenen der Projektorganisation - das Portfolio, das Programm und das Projekt. Scrum Teams haben die Möglichkeit, den Scrum Guidance Body wenn nötig um Rat zu fragen.

Bei der Erstellung von Epics kann sich die die Fachkenntnis aus dem Scrum Guidance Body auf die dokumentierten Regeln und Regularien oder auf Standards und besten Methoden für die Entwicklung von Epics beziehen. Es kann darüber hinaus auch ein Team von Experten dem Product Owner bei der Entwicklung von Epics zur Seite stehen. Dieses Team kann Geschäftsanalysten, führende Architekten, erfahrene Entwickler, Scrum Experten und andere erfahrene Personen umfassen. Die Expertengruppe ist üblicherweise nicht das selbe Team, das schließlich am eigentlichen Projekt arbeiten wird, da sie von Projekt zu Projekt wandern, wenn sich diese in der 'Verkaufsphase' oder der 'Phase Null' mit den Kunden oder Anwendern befinden.

8.4.2.8 Scrum Projekttool

Siehe Abschnitt 2.5.3.1

8.4.3 Outputs

8.4.3.1 Epic(s)*

Epics werden in den frühen Phasen des Projekts niedergeschrieben, wenn die meisten Anforderungen high-level Funktionalitäten sind, oder Produktbeschreibungen nur sehr allgemein definiert sind. Epics existieren als große, nicht verfeinerte User Storys im priorisierten Produkt Backlog. Epics helfen dem Product Owner und den relevanten Business Stakeholdern bei der Planung von Releases, der Priorisierung von High-Level-Anforderungen und der Erstellung einer allgemeinen Roadmap für das Projekt.

Wenn der Product Owner mehr Klarheit über die Benutzeranforderungen erlangt, werden diese Epics dann in kleinere, detailliertere User Storys aufgeteilt. Epics enthalten also eine oder mehrere User Storys. Bei diesen kleineren User Storys handelt es sich in der Regel um einfache, kurze und leicht zu implementierende Funktionalitäten oder Aufgabenblöcke, die in einem Sprint abgeschlossen werden können. User Storys sollten vom Product Owner definiert werden und müssen der Definition von Ready entsprechen und sollten abgeschätzt werden, bevor sie im priorisierten Produkt Backlog zur Aufnahme in einen kommenden Sprint anstehen.

Sobald diese Epics im Priorisierten Produkt Backlog für die Durchführung in einem anstehenden Sprint auftauchen, werden sie in kleinere, mehr granulare User Storys heruntergebrochen. Diese kleineren User Storys sind üblicherweise einfache, kurze, und leicht durchzuführende Funktionalitäten oder Aufgabenblöcke, welche in einem Sprint zu realisieren sind.

8.4.3.2 Personas*

Personas sind sehr detaillierte fiktionale Figuren, welche repräsentativ für die Mehrheit der Anwender oder anderer Business Stakeholdern, welche das Endprodukt eventuell nicht direkt nutzen, stehen. Personas werden erschaffen, um die Bedürfnisse der beabsichtigten Anwenderbasis zu ermitteln. Die Erstellung von spezifischen Personas kann dem Team dabei helfen, die Anwender und ihre Anforderungen und Ziele besser zu verstehen. Basierend auf einer Persona kann der Product Owner das priorisierte Produkt Backlog effektiver erstellen.

Erstellen einer Persona: Beinhaltet die Zuordnung eines fiktiven Namens und vorzugsweise eines Bildes (beispielsweise ein Stock Image) zur Figur. Die Persona umfasst sehr spezifische Attribute, wie beispielsweise Alter, Geschlecht, Ausbildung, Umfeld, Interessen, und Ziele. Ein Zitat, welches die Anforderung der Persona wiedergibt, kann ebenfalls enthalten sein. Im Folgenden ist ein Beispiel für eine Persona einer Reise-Webseite angegeben.

Beispiel:

Vanessa ist 39 Jahre alt und lebt in San Francisco. Nach einer sehr erfolgreichen Karriere als Anwältin lebt sie ihre Leidenschaft für das Reisen aus. Sie bevorzugt es, bei der Auswahl von Flügen und Unterkünften Optionen zu haben, so dass sie die beste und preiswerteste auswählen kann. Sie reagiert frustriert auf langsame und unübersichtliche Webseiten.

8.4.3.3 Genehmigte Change Requests

Change Requests können durch den Product Owner während der Entwicklung von Epic(s) genehmigt werden, dies mitunter anhand von Vorschlägen durch die relevanten Business Stakeholdern. Solche Änderungen werden als genehmigte Changes kategorisiert, können priorisiert und in zukünftigen Sprints implementiert werden.

Change Requests und Genehmigte Change Requests werden in den Abschnitten 6.3.1, 6.4.2.1 und 6.6 besprochen.

8.4.3.4 Identifizierte Risiken

Bei der Entwicklung von Epics kann es sein, dass neue Risiken identifiziert werden; diese identifizierten Risiken stellen ein wichtiges Ergebnis dieser Phase dar. Die Risiken tragen zur Entwicklung des priorisierten Produkt Backlogs bei (welches auch als Risikobasiertes Produkt Backlog bezeichnet werden kann).

Die Risikoidentifikation wird im Abschnitt 7.4.1 beschrieben.

8.5 Priorisiertes Produkt Backlog Erstellen

In diesem Prozess werden die Epics verfeinert, ausgearbeitet und, was am wichtigsten ist, nach ihrem jeweiligen Geschäftswert priorisiert, um ein priorisiertes Product Backlog für das Projekt zu erstellen. Zusätzlich legen der Product Owner und das Scrum Team auf der Grundlage der Empfehlungen des Scrum Guidance Body die Done Kriterien für das Projekt fest.

Abbildung 8-12 zeigt um so Inputs, Hilfsmittel und Outputs für *Priorisiertes Produkt Backlog Erstellen* Prozess.

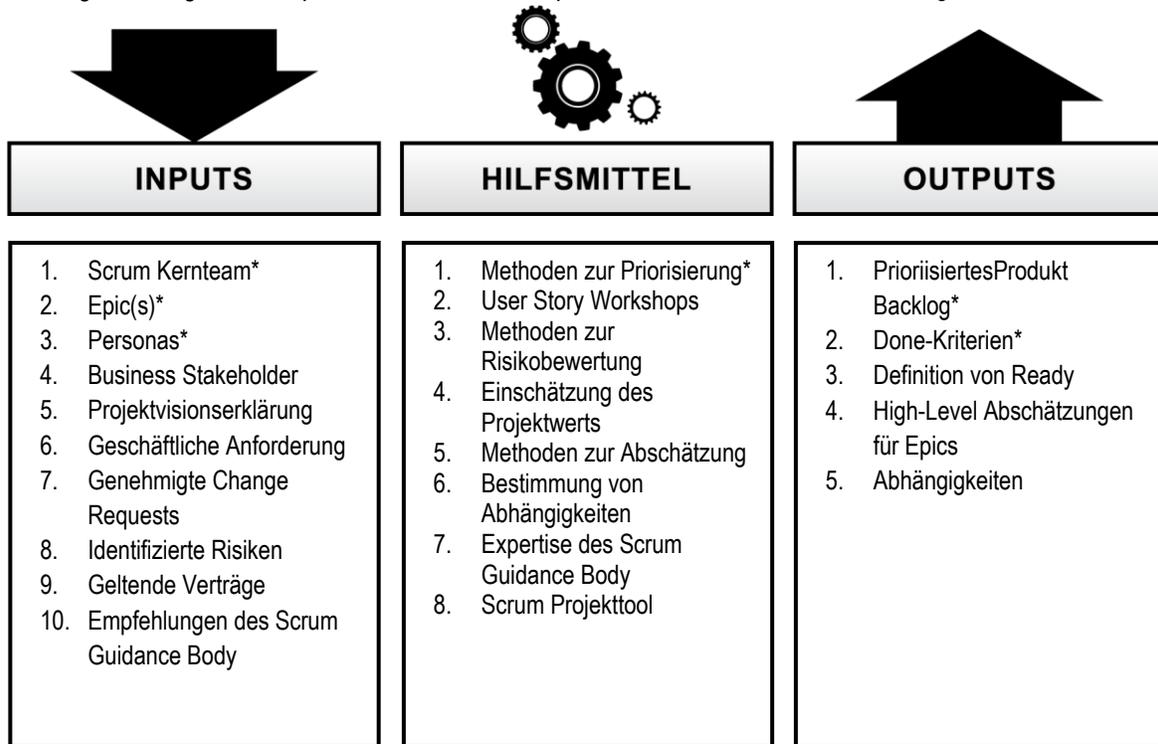


Abbildung 8-12: Priorisiertes Produkt Backlog Erstellen–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

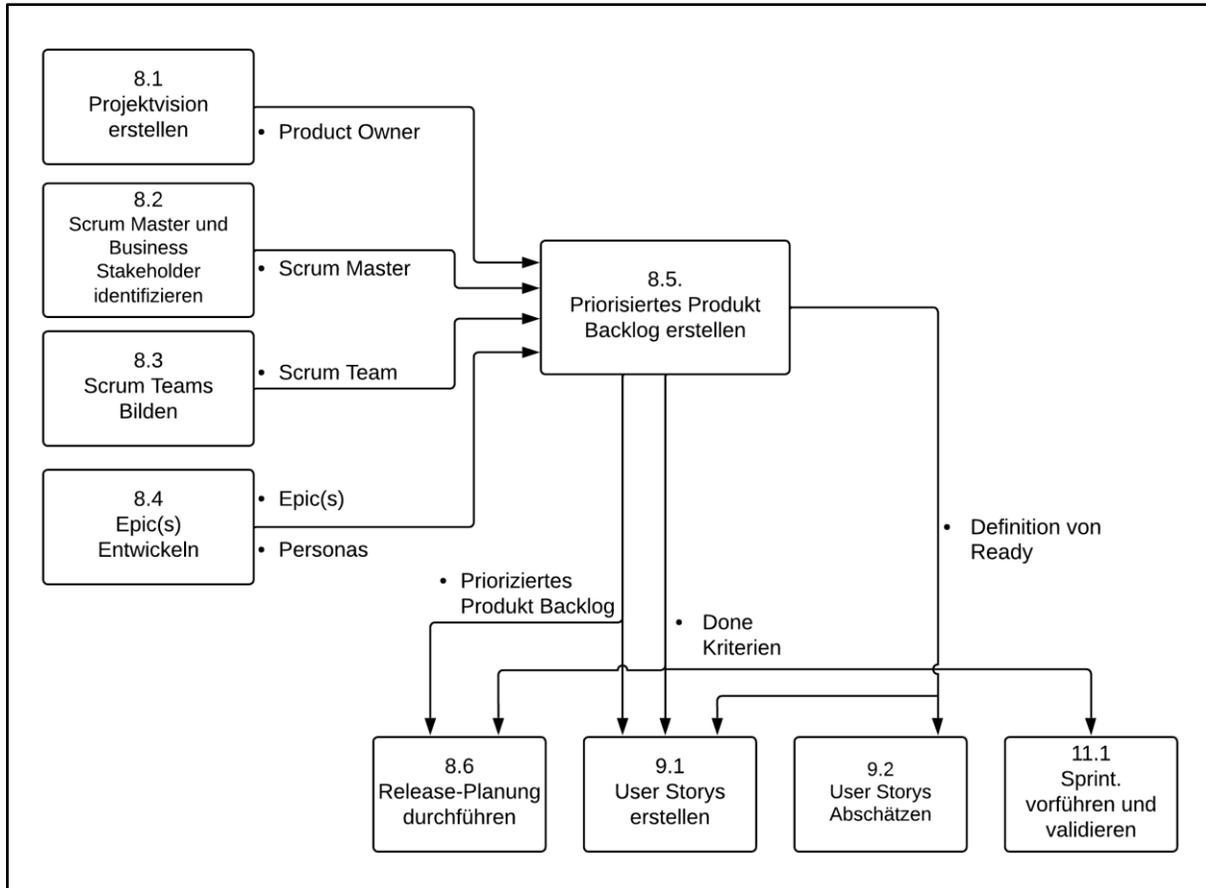


Abbildung 8-13: Erstellen eines priorisierten Produkt Backlogs: Datenflussdiagramm

8.5.1 Inputs

8.5.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

8.5.1.2 Epic(s)*

Siehe Abschnitt 8.4.3.1.

8.5.1.3 Personas*

Siehe Abschnitt 8.4.3.2.

8.5.1.4 Business Stakeholdern

Siehe Abschnitt 8.2.3.2.

8.5.1.5 Projektvisionserklärung

Siehe Abschnitt 8.1.3.2.

8.5.1.6 Programm Produkt Backlog

Siehe Abschnitt 8.1.1.6.

8.5.1.7 Geschäftliche Anforderungen

Die Summe aller Erkenntnisse, welche durch verschiedene Mittel wie die Anwender- und Kundeninterviews, Fragebögen, JAD-Sitzungen, Lückenanalysen, SWOT-Analysen und anderen Meetings erlangt wurden, hilft dabei, ein besseres Verständnis für die geschäftlichen Anforderungen zu erhalten und das Priorisierte Produkt Backlog zu entwickeln.

8.5.1.8 Genehmigte Change Requests

Siehe Abschnitt 8.4.3.3.

8.5.1.9 Identifizierte Risiken

Siehe Abschnitt 8.4.3.4.

8.5.1.10 Geltende Verträge

Siehe Abschnitt 8.4.1.9.

8.5.1.11 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11.

Bei der Erstellung des priorisierten Produkt Backlogs können die Empfehlungen des Scrum Guidance Body Informationen über Regeln, Verordnungen, Standards und beste Methoden für die Entwicklung eines priorisierten Produkt Backlogs enthalten.

8.5.2 Hilfsmittel

8.5.2.1 Methoden zur Priorisierung von User Storys*

Einige Methoden, welche zur Priorisierung der User Storys oder Anforderungen im priorisierten Produkt Backlog auf Basis des Geschäftswerts angewendet werden, sind im folgenden beschrieben:

- **Das MoSCoW-Priorisierungsschema**—Das MoSCoW-Priorisierungsschema leitet seinen Namen von den ersten Buchstaben der Phrasen „Must have“ (Muss), „Should have“ (Sollte), „Could have“ (Kann/Könnte) und „Would like to have, but not at this time“ (Würde [ich] gern haben, aber nicht jetzt) ab. Diese Priorisierungsmethode ist generell effektiver als einfache Schemata. Die Phrasen stehen in einer absteigenden Reihenfolge im Bezug auf Priorität: „Must Have“ User Storys sind dabei diejenigen, ohne welche das Projekt keinen Wert hätte, während „Would like to Have“ solche sind, welche zwar positiv wären, aber nicht zwingend nötig sind.
- **Paarweiser Vergleich**—Bei dieser Methode wird eine Liste aller User Storys im priorisierten Produkt Backlog erstellt. Als nächstes werden die User Storys individuell und eine nach der anderen mit den anderen bereits behandelten Storys verglichen.

Der Vergleich erfolgt immer zwischen zwei User Storys, wobei eine Entscheidung getroffen wird, welche der beiden wichtiger ist. Auf diese Weise kann eine priorisierte Liste von User Storys generiert werden.

- **100-Punkte-Methode**—Die 100-Punkte-Methode wurde von Dean Leffingwell und Don Widrig (2003) entwickelt. Dabei werden den Kunden 100 Punkte gegeben, welche sie zur Wahl der wichtigsten User Storys nutzen können. Das Ziel ist es, die User Storys mit höherer Priorität im Vergleich zu anderen stärker zu gewichten. Jedes Mitglied der Gruppe verteilt die Punkte auf die verschiedenen User Storys, und gibt denjenigen, welche sie als wichtiger erachten, entsprechend mehr Punkte. Am Ende des Abstimmvorgangs wird die Priorisierung anhand der totalen Punktzahl jeder User Story durchgeführt.
- **Kano-Analyse**

Siehe Abschnitt 4.5.2

8.5.2.2 User Story Workshops

Siehe Abschnitt 8.4.2.2.

8.5.2.3 Methoden der Risikobewertung

Siehe Abschnitt 7.4.2.1.

8.5.2.4 Abschätzung des Projektwerts

Siehe Abschnitt 4.5.1.

8.5.2.5 Abschätzungsmethoden

Es kann sinnvoll sein, einige sehr grobe Abschätzungen für Epics zu formulieren, da diese Schätzungen dem Product Owner helfen können, die Releases angemessen zu planen, und auch bei der Priorisierung helfen können. Da die in dieser Phase geplanten Epics jedoch in der Regel noch nicht in genau definierte User Storys heruntergebrochen sind und auch die Möglichkeit besteht, dass einige der ursprünglich entwickelten Epics nicht implementiert werden, sollte das Team zu diesem Zeitpunkt nicht zu viel Zeit auf die Abschätzung der Epics verwenden.

Dieses Tool wird im Prozess User Storys abschätzen (wie in Abschnitt 9.2.2.1 beschrieben) anders eingesetzt, bei dem das Scrum Team sehr stark in die Abschätzung von gut definierten User Storys eingebunden ist. Der Product Owner, der Scrum Master und die relevanten Business Stakeholdern sollten sich darüber im Klaren sein, dass alle High-Level-Abschätzungen aus diesem Prozess nur als Anhaltspunkte dienen und dass es erhebliche Abweichungen von den Schätzungen des Scrum Teams für die User Storys geben kann, sobald die User Storys richtig definiert und geschätzt sind.

Einige spezifische Tools, die zur Erstellung von High-Level-Abschätzungen für Epics verwendet werden können, sind:

- Bereits vorhandene Epic-Schätzungen - Wenn ähnliche Epics in der Vergangenheit für dasselbe Projekt oder andere verwandte Projekte entwickelt wurden, können diese früheren Schätzungen zur Herleitung von Abschätzungen in diesem Prozess verwendet werden.
- Expertenmeinungen - Experten, die in der Vergangenheit ähnliche Projektfunktionalitäten implementiert haben, können unter Umständen grobe Aufwandsschätzungen für Epics abgeben.
- 5-10-Minuten-Schätzungen - Die relevanten Teammitglieder können einen Überblick über die allgemeinen Anforderungen für jedes Epic erhalten und gebeten werden, innerhalb von fünf bis zehn Minuten eine grobe Abschätzung (z. B. in Stunden, Tagen oder Wochen) abzugeben.

Diese Schätzungsmethode sollte in Betracht gezogen werden, wenn nur eine kleine Anzahl von Epics geschätzt werden muss, da sie sonst zu zeitaufwändig werden könnte.

- Vergleichende Abschätzung (Affinitätsschätzung) Die Vergleichende Abschätzung (z. B. T-Shirt-Sizing) ist eine Technik, mit der schnell eine Abschätzung der Zeit für eine große Anzahl von Epics erstellt werden kann. Dies wird in Abschnitt 9.2.2.1 ausführlich beschrieben.

8.5.2.6 Bestimmung der Abhängigkeiten

Die ordnungsgemäße Dokumentation von Abhängigkeiten hilft dem Scrum Team, die relative Reihenfolge zu bestimmen, in der Epics (und User Storys) ausgeführt werden sollten, um die zu Liefernden Ergebnisse des Sprints zu erstellen. Abhängigkeiten verdeutlichen die Beziehung und Interaktion zwischen Epics (und zwischen User Storys), sowohl innerhalb des Scrum Teams, das an einem bestimmten Sprint arbeitet, als auch mit anderen Scrum Teams, die an dem Projekt arbeiten. Bei der Erstellung des priorisierten Product Backlogs identifiziert der Product Owner alle Abhängigkeiten und Beziehungen zwischen Epics (und zwischen User Storys), einschließlich aller technischen Abhängigkeiten und Abhängigkeiten in Bezug auf die Verfügbarkeit von Mitarbeitern, da diese Abhängigkeiten die Reihenfolge und Priorität der im Projekt zu erledigenden Arbeit beeinflussen werden.

Abhängigkeiten können obligatorisch oder diskretionär, intern oder extern oder eine Kombination dieser Faktoren sein. Eine Abhängigkeit kann beispielsweise sowohl obligatorisch als auch projektextern sein.

Im Folgenden werden die einzelnen Arten von Abhängigkeiten beschrieben.

- **Obligatorische Abhängigkeiten** - Obligatorische Abhängigkeiten sind Abhängigkeiten, die die Reihenfolge der im Projekt zu erledigenden Arbeiten vorgeben und daher bei der anfänglichen Erstellung des priorisierten Produkt-Backlogs zu berücksichtigen sind. Diese Abhängigkeiten können aufgrund der Art der Arbeit zwingend erforderlich sein (z. B. eine physische Einschränkung der Arbeitsreihenfolge) oder aufgrund vertraglicher Verpflichtungen oder rechtlicher Anforderungen bestehen. Obligatorische Abhängigkeiten werden auch als harte Logik bezeichnet.
- **Diskretionäre Abhängigkeiten** - Diskretionäre Abhängigkeiten sind Abhängigkeiten, die aufgrund früherer Erfahrungen oder bewährter Verfahren in einem bestimmten Bereich oder einer bestimmten Domäne in den Workflow eingefügt werden. So kann das Team beispielsweise beschließen, ein Epic (oder eine User Story) vor einem anderen abzuschließen, weil sich dieser Arbeitsablauf bei früheren Projekten bewährt hat.
- **Externe Abhängigkeiten** - Externe Abhängigkeiten sind Abhängigkeiten, die sich auf Aktivitäten oder Faktoren beziehen, die außerhalb des Umfangs der vom Scrum-Team auszuführenden Arbeit liegen, aber für den Abschluss einer Aufgabe oder die Erstellung eines zu Liefernden Ergebnisses im Projekt erforderlich sind. Externe Abhängigkeiten liegen in der Regel außerhalb der Kontrolle des Scrum-Teams und können daher ein größeres Risiko für ein Projekt darstellen.
- **Interne Abhängigkeiten** - Interne Abhängigkeiten sind die Faktoren innerhalb des Projekts, die sich auf die Abfolge der zu erledigenden Arbeiten auswirken. Diese Faktoren liegen normalerweise unter der Kontrolle des Scrum-Teams.

Es gibt verschiedene Möglichkeiten, Projektaufgaben und ihre Abhängigkeiten zu identifizieren, zu definieren und darzustellen.

8.5.2.7 Fachkenntnis des Scrum Guidance Body

Siehe Abschnitt 8.4.2.7

Während *Erstellen des Priorisierten Produkt Backlog* kann sich die Fachkenntnis des Scrum Guidance Body auf dokumentierte Regeln und Verordnungen oder Standards und beste Methoden zur Erstellung von Epics beziehen. Es kann darüber hinaus auch ein Team von Experten dem Product Owner bei der Entwicklung von Epics zur Seite stehen. Dieses Team kann Geschäftsanalysten, führende Architekten, erfahrene Entwickler, Scrum Experten und andere erfahrene Personen umfassen. Die Expertengruppe ist üblicherweise nicht dasselbe Team, das schließlich am eigentlichen Projekt arbeiten wird, da sie von Projekt zu Projekt wandern, wenn sich diese in der 'Verkaufsphase' oder der 'Phase Null' mit den Kunden oder Anwendern befinden.

8.5.2.8 Scrum Projekttool

Siehe Abschnitt 2.5.3.1

8.5.3 Outputs

8.5.3.1 Priorisiertes Produkt Backlog*

Der Product Owner erstellt das priorisierte Produkt Backlog, welches eine priorisierte Liste von geschäftlichen und projektbezogenen Anforderungen in der schriftlichen Form der Epics beinhaltet, welche high-level User Storys darstellen. Das priorisierte Produkt Backlog basiert auf drei primären Faktoren: Wert, Risiko und Ungewissheit, und Abhängigkeiten. Es wird auch als risikobezogenes Produkt Backlog bezeichnet, da es identifizierte und bewertete Risiken, die mit dem Projekt in Verbindung stehen, beinhaltet. Es umfasst darüber hinaus alle genehmigten Änderungen, welche entsprechend im priorisierten Produkt Backlog priorisiert werden können (siehe Abschnitt 6.3.1).

- **Wert**—Es ist die Aufgabe des Product Owners, die Lieferung derjenigen Anforderungen mit dem höchsten Geschäftswert zuerst abzusichern. Selbst eine extrem wertvolle Anforderung kann mitunter nicht Teil des ersten Release sein, wenn andere mit höherem Wert in hinlänglichem Umfang für ein erstes Release vorhanden sind.
- **Risiko und Ungewissheit**—Je mehr Ungewissheit existiert, desto riskanter ist das Projekt. Daher ist es wichtig, dass die riskantere Anforderung im priorisierten Produkt Backlog eine höhere Priorität erhält.

Anforderungen mit einem höheren Risikolevel werden darüber hinaus auch risikomindernde Handlungen erfordern. Wenn diese risikomindernden Handlungen gegen das Backlog priorisiert werden, entsteht dadurch ein risikobezogenes Produkt Backlog. Die Behandlung von Risiken zu Beginn des Projekts garantiert nicht den Erfolg des Projekts, aber es erhöht die Fähigkeit des Teams, mit dem Risiko umzugehen. Siehe Abschnitt 7.4.3.

- **Abhängigkeiten**—In den meisten Projekten gibt es Abhängigkeiten zwischen einigen Epics oder User Storys. Diese Abhängigkeiten müssen bei der Erstellung eines priorisierten Produkt Backlogs berücksichtigt werden. Funktionale Anforderungen hängen oft von anderen funktionalen oder gar non-funktionalen Anforderungen ab. Diese Abhängigkeiten können beeinflussen, wie die Epics (und User Storys) in einem priorisierten Produkt Backlog priorisiert werden. Zwei der verbreitetsten Methoden, diese Abhängigkeiten zu lösen, sind das Aufteilen eines Epics (oder einer einzelnen Story) in mehrere Teile oder das Kombinieren von voneinander abhängigen Bestandteilen.
- **Abschätzungen**—High-Level Abschätzungen für Epic(s) sind ebenfalls im priorisierten Produkt Backlog vorhanden.

Es ist wichtig zu beachten, dass sich die Priorisierung von Epics von der Priorisierung der zugrunde liegenden User Storys unterscheiden kann. Selbst wenn beispielsweise ein Epic als hoch priorisiert eingestuft wird, können einige der im Epic enthaltenen User Storys als niedrig priorisiert eingestuft werden, während andere User Storys im Epic als hoch priorisiert eingestuft werden können.

8.5.3.2 Done-Kriterien*

Done-Kriterien sind eine Gruppe von Regeln, welche auf alle User Storys anwendbar sind. Eine klare Definition von Erledigt/Done ist essentiell, da dies Mehrdeutigkeiten bei den Anforderungen beseitigt und dem Team hilft, den vorgeschriebenen Qualitätsnormen zu entsprechen. Diese klare Definition wird genutzt, um die Done-Kriterien zu erstellen, welche ein Ergebnis der Erstellung des priorisierten Produkt Backlogs sind. Eine User Story wird als Erledigt/Done erachtet, wenn Sie dem Product Owner vorgeführt und durch diesen, der sie auf Basis der Done-Kriterien und der User-Story Abnahmekriterien bewertet, akzeptiert ist.

Beispiel für Done-Kriterien:

Projekt: Entwicklung von neuen Varianten eines beliebten Sportwagens der LRA Ltd.

Done-Kriterien:

- Das Design wird von der Abteilung für Technische Exzellenz abgesegnet.
- Der Prototyp besteht alle Windkanaltests, welche von der Aerodynamik-Abteilung vorgeschrieben werden.
- Das Design wird von der Abteilung für geistiges Eigentum freigegeben.
- Die Erwartungen an die Sicherheit des Designs werden durch den Designsicherheitsbericht der Sicherheitsabteilung bestätigt.
- Der Report fuer die Kostenabschaetzung ist durch die Finanzabteilung abgesegnet.

8.5.3.3 Definition von Bereit (Definition of Ready)*

Die Definition von Bereit (Definition of Ready) definiert die Kriterien, die eine User Story erfüllen muss, bevor sie für die Abschätzung und Aufnahme in einen Sprint in Frage kommt. Die Definition von Bereit legt die Verantwortung auf den Product Owner, die Anforderungen für jede User Story angemessen zu definieren. Ohne korrekt definierte Anforderungen ist es unmöglich, verlässliche Schätzungen zu erhalten, und das Scrum-Team wird nicht in der Lage sein, die erforderliche Projektarbeit effektiv zu erledigen.

8.5.3.4 High-Level-Abschätzungen für Epics*

High-Level-Abschätzungen für Epics können vom Product Owner unter Verwendung verschiedener Abschätzungsmethoden initiiert werden (wie in Abschnitt 8.5.2.5 beschrieben). Diese Abschätzungen helfen dem Product Owner, eine ungefähre Vorstellung davon zu bekommen, wie viel Zeit und Aufwand die Fertigstellung jedes Epics erfordert, was wiederum bei der Priorisierung der Epics im priorisierten Product Backlog und bei der Planung von Projekt-Releases hilft.

8.5.3.5 Abhängigkeiten*

Abhängigkeiten beschreiben die Beziehung und Interaktion zwischen verschiedenen Epics (oder User Stories) in einem Projekt. Abhängigkeiten können als obligatorisch oder diskretionär, intern oder extern oder als eine Kombination davon klassifiziert werden (wie in Abschnitt 8.5.2.6 beschrieben). Abhängigkeiten wirken sich auf die relative Reihenfolge aus, in der Epics (und User Stories) ausgeführt werden, um die zu Liefernden Ergebnisse des Sprints zu erstellen, und wirken sich daher auf ihre im priorisierten Product Backlog dokumentierte Priorität aus.

8.6 Release-Planung durchführen

In diesem Prozess entwickelt der Product Owner mit Unterstützung des Scrum-Teams die anfängliche Release-Planung, die den Business Stakeholdern mitgeteilt und mit ihnen geteilt wird. Es wird davon ausgegangen, dass die iterative Natur von Scrum zukünftige Anpassungen des Release Plans erforderlich machen kann. Die Länge der einzelnen Sprints wird ebenfalls in diesem Prozess festgelegt.

Abbildung 8-14 zeigt Inputs, Hilfsmittel und Outputs für *Release-Planung Durchführen* Prozess.

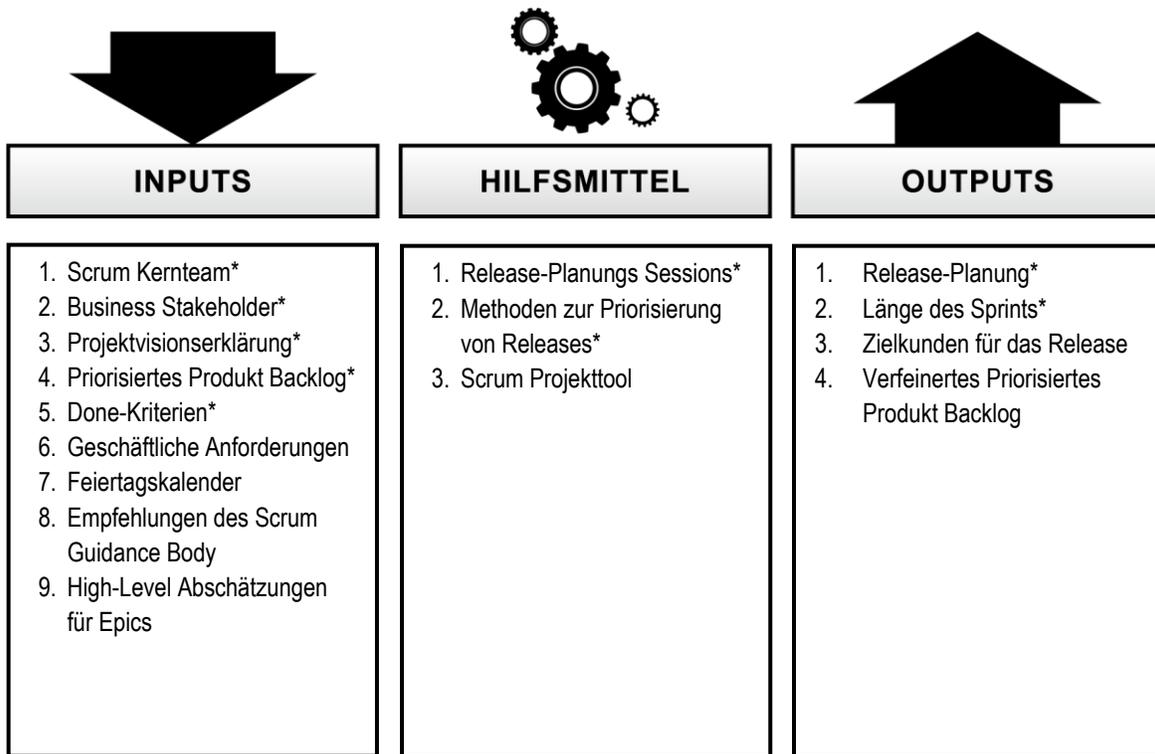


Abbildung 8-14: Release-Planung durchführen-Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

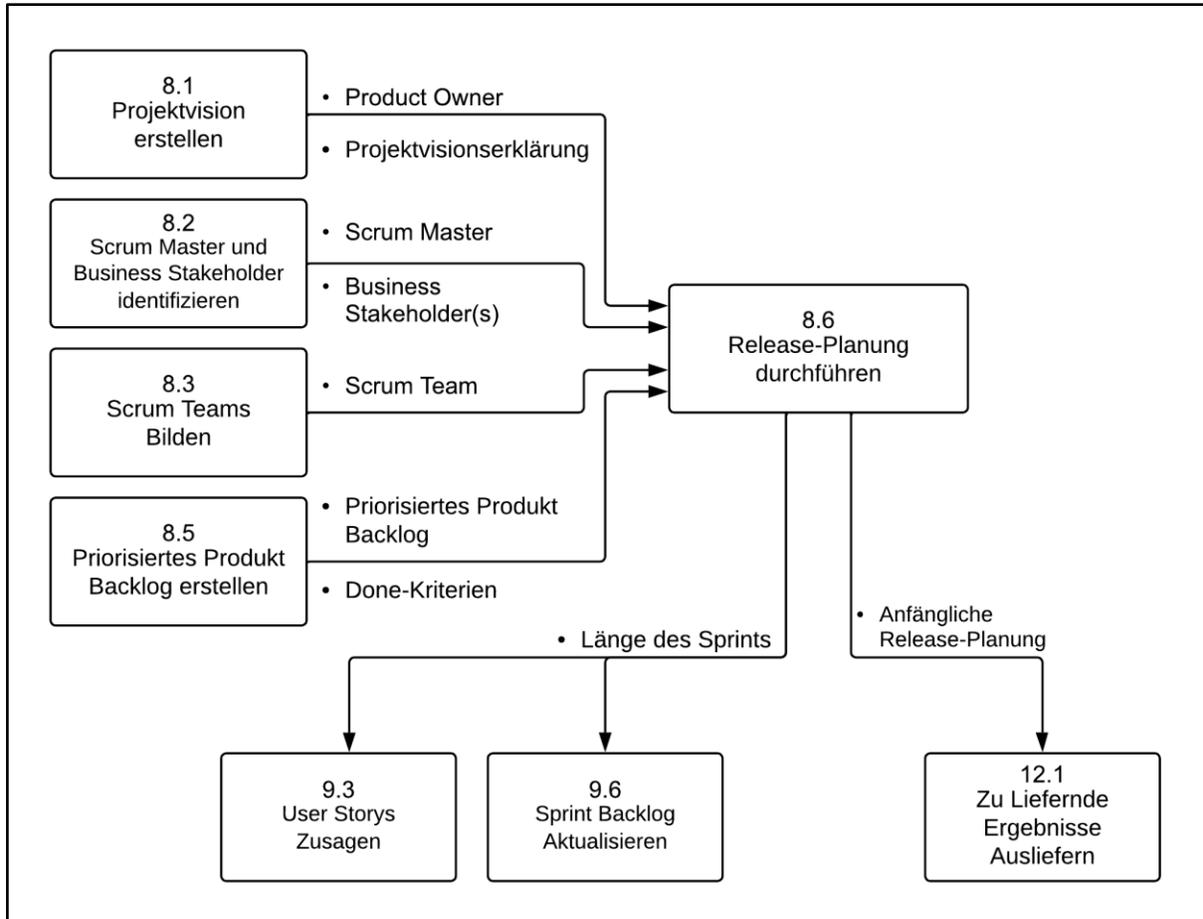


Abbildung 8-15: Release-Planung durchführen–Datenflussdiagramm

8.6.1 Inputs

8.6.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

8.6.1.2 Business Stakeholdern*

Siehe Abschnitt 8.2.3.2.

8.6.1.3 Projektvisionserklärung*

Siehe Abschnitt 8.1.3.2.

8.6.1.4 Priorisiertes Produkt Backlog*

Siehe Abschnitt 8.5.3.1.

8.6.1.5 Done-Kriterien*

Siehe Abschnitt 8.5.3.2.

8.6.1.6 Geschäftliche Anforderungen

Siehe Abschnitt 8.5.1.7.

8.6.1.7 Feiertagskalender

Es ist wichtig, dass das Scrum Team den Überblick über Schlüsseldaten und die Verfügbarkeit aller Teammitglieder behält. Das kann durch einen gemeinsamen Kalender, welcher Informationen wie offizielle Feiertage, Urlaub, Reisepläne, Ereignisse etc. bereitstellt, erreicht werden. Dieser Kalender kann dem Team bei der Planung und Durchführung von Sprints helfen.

8.6.1.8 Empfehlungen des Scrum Guidance Body

Bei der Durchführung der Release-Planung können sich Empfehlungen des Scrum Guidance Body auf Regeln, Verordnungen, Standards und beste Methoden für die Entwicklung eines Release-Plans beziehen. Der Guidance Body kann dabei die beste Instanz für die Entwicklung von Richtlinien im Bezug auf Geschäftswert, Erwartungen an das Release, Qualität und Sicherheit sein.

8.6.1.9 High-Level-Abschätzungen für Epics*

In Release-Planungs-Sessions werden High-Level-Abschätzungen für Epics verwendet, um Releases entweder nach einem zeit- oder einem funktionsorientierten Ansatz zu planen. Weitere Informationen zu diesen Abschätzungen finden Sie in den Abschnitten 8.5.2.5 und 8.5.3.4.

8.6.2 Hilfsmittel

8.6.2.1 Release Planungs-Sessions*

Release Planungs-Sessions werden zur Entwicklung eines Release-Plans durchgeführt. Der Plan definiert, wann verschiedene Gruppen von nutzbaren Funktionalitäten oder Produkte an den Kunden weitergegeben werden. Bei Scrum ist eines der wesentlichen Ziele eines Release-Planungs-Meetings, dem Scrum Team einen Überblick über die Freigaben und den Lieferzeitplan für die Produkte, die sie entwickeln, zu ermöglichen, so dass sie sich auf die Erwartungen des Product Owners und der relevanten Business Stakeholdern (insbesondere der Sponsoren) einrichten können.

Viele Organisationen haben eine Strategie für das Release eines Produkts. Einige Organisationen bevorzugen eine kontinuierliche Auslieferung, bei der ein Release nach der Entwicklung von spezifischen nutzbaren Funktionalitäten erfolgt. Andere Organisationen bevorzugen eine phasenweise Auslieferung, bei der Releases in vor-definierten Intervallen erfolgt. Je nach der Strategie der Organisation können Release-Planungs-Sessions in Projekten von Funktionalität geprägt sein, bei der das Ziel ist das Produkt zu liefern, nachdem eine vorherbestimmte Reihe von Funktionalitäten entwickelt wurde, oder die Planung ist datums-geprägt, so dass das Release zu einem vorherbestimmten Zeitpunkt erfolgt.

Da der Scrum-Rahmen eine informations-basierte, iterative Entscheidungsfindung gegen über der detaillierten Planung im Vorfeld, wie sie vom traditionellen Projektmanagement im Wasserfallstil betrieben wird, bevorzugt, müssen Planungs-Sessions nicht einen detaillierten Release-Plan für das gesamte Projekt entwickeln. Der Release-Plan kann beständig aktualisiert werden, wenn relevante Informationen bekannt werden.

8.6.2.2 Methoden zur Priorisierung von Releases*

Die Methoden zur Priorisierung von Releases werden genutzt, um einen Release-Plan zu entwickeln. Diese Methoden sind branchen- und organisationsspezifisch und werden üblicherweise durch die Geschäftsleitung der Organisation bestimmt.

8.6.3 Outputs

8.6.3.1 Release-Plan*

Ein Release Plan ist eines der Schlüsselergebnisse bei der Durchführung der Release-Planung. Ein Release-Plan gibt an, auf welche Weise die zu liefernden Ergebnisse an den Kunden weitergegeben werden sollen, zusammen mit den geplanten Zeiträumen und Daten der Veröffentlichung/Freigabe. Ein Release umfasst in der Regel eine Gruppe von User Storys im priorisierten Product Backlog, die als Teil des Releases fertiggestellt und gemeinsam ausgeliefert werden sollten.

Es ist jedoch wichtig zu beachten, dass die User Storys in einem Release nicht immer die vollständige Funktionalität von Epics bieten. Einige Epics werden möglicherweise nur teilweise fertiggestellt, da nicht alle User Storys in einem Epic notwendigerweise eine ausreichend hohe Priorität haben, um Teil dieses spezifischen Releases zu sein.

Nicht nach jedem Sprint muss ein Release eingeplant sein. Mitunter kann ein Release auch nach dem Abschluss einer Gruppe von mehreren Sprints eingeplant werden. Je nach der Strategie der Organisation können Release-Planungs-Sessions in Projekten von Funktionalität geprägt sein, bei der das Ziel ist, das Produkt zu liefern, nachdem eine vorherbestimmte Reihe von Funktionalitäten entwickelt wurde, oder die Planung ist datumsgeprägt, so dass das Release zu einem vorherbestimmten Zeitpunkt erfolgt. Das zu liefernde Ergebnis sollte freigegeben werden, wenn es einen ausreichenden Geschäftswert für den Kunden bereithält.

Während des Prozesses Release Planung durchführen kann es sinnvoll sein, die Sprint-Länge und die Annahmen für die Team-Geschwindigkeit zu berücksichtigen, um einen besseren Release-Plan abzuleiten.

8.6.3.2 Länge des Sprints*

Basierend auf den verschiedenen Vorgaben, darunter die geschäftlichen Anforderungen und der Release-Plan, entscheiden der Product Owner und das Scrum Team die Länge der Sprints im Projekt. Sobald sie bestimmt ist, bleibt die Länge der Sprints oft während des gesamten Projekts gleich.

Dennoch kann die Länge der Sprints geändert werden, wenn und wann es der Product Owner und das Scrum Team als erforderlich erachten. Zu Beginn des Projekts kann es sein, dass sie noch experimentieren, um die beste Sprintlänge zu finden. Im späteren Verlauf des Projekts bedeutet eine Änderung der Länge des Sprints normalerweise, dass sie auf Grund von Verbesserungen der Projektumgebung verkürzt werden kann.

Um den größtmöglichen Nutzen aus einem Scrum-Projekt zu ziehen und ein Höchstmaß an Flexibilität für Änderungen zu gewährleisten, sollte die Dauer eines Sprints so kurz wie möglich sein. Ein Sprint ist typischerweise Time-boxed mit einer Dauer von einer bis vier Wochen. Bei den meisten Scrum-Projekten haben Sprints in der Regel eine Time-Box-Dauer von zwei oder drei Wochen. Bei Projekten mit sehr stabilen Anforderungen können Sprints jedoch bis zu sechs Wochen dauern, wenn dies gerechtfertigt ist.

Da während eines Sprints keine Änderungen erlaubt sind, können die Auswirkungen und die Häufigkeit von Änderungen einen Einfluss auf die Entscheidung über die Länge des Sprints haben. Die Auswirkungen einer zu erwartenden Änderung der Sprintlänge wird in Abschnitt 6.5.1 beschrieben.

8.6.3.3 Zielkunden für das Release

Nicht jedes Release richtet sich an alle Business Stakeholdern oder Anwender. Die Business Stakeholdern können sich mitunter entscheiden, bestimmte Releases auf einen Teil der Anwender zu beschränken. Der Release-Plan sollte die Zielkunden für das Release definieren.

8.6.3.4 Verfeinertes priorisiertes Produkt Backlog

Das zuvor entwickelte priorisierte Produkt Backlog kann in diesem Prozess verbessert werden. Es kann zusätzliche Klarheit über User Storys im priorisierten Produkt Backlog bestehen, nachdem das Scrum Kernteam zusammen mit den Business Stakeholdern die Release-Planungs-Sitzungen abgehalten hat.

8.7 Datenflussdiagramm der Initiierungs-Phase

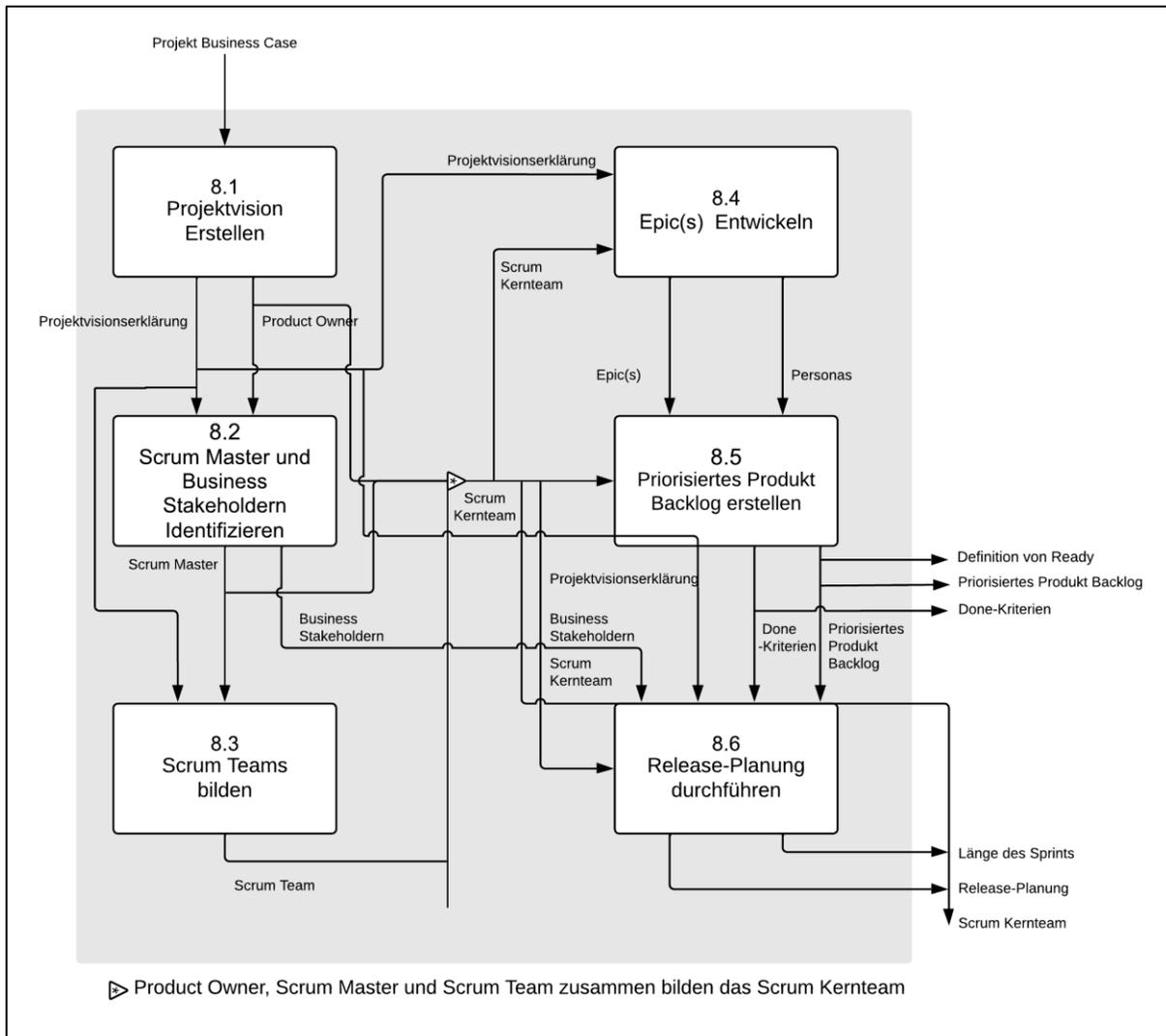


Abbildung 8-16: Initiierungsphase-Datenflussdiagramm

9. PLANEN UND ABSCHÄTZEN

Die Planungs- und Abschätzungsphase besteht aus Prozessen, die sich mit der Planung und Abschätzung von Aufgaben beschäftigen. Zu diesen Prozessen gehören, *User Storys erstellen*, *User Storys abschätzen*, *User Storys zusagen*, *Aufgaben identifizieren*, *Aufgaben abschätzen*, und *Sprint Backlog aktualisieren*.

Planen und Abschätzen, wie im *Scrum Body of Knowledge (SBOK®)* definiert, ist anwendbar auf:

- Portfolios, Programme, und/oder Projekte in jeder Branche
- Produkte, Dienstleistungen, oder jegliche an Business Stakeholdern zu liefernde Ergebnisse
- Projekte jeglicher Größe oder Komplexität

Der Begriff "Produkt" in dieser Wissenssammlung kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann auf jedes Projekt in jeder Branche effektiv angewendet werden — von kleinen Projekten oder Teams mit lediglich sechs Team-Mitgliedern bis hin zu großen, komplexen Projekten mit mehreren Hundert Team-Mitgliedern.

Um die beste Anwendung des Scrum Rahmens zu ermöglichen, kennzeichnet dieses Kapitel Inputs, Hilfsmittel und Outputs für jeden Prozess entweder als „verbindlich“ oder „optional“. Inputs, Hilfsmittel und Outputs, die mit einem Sternchen (*) gekennzeichnet sind, sind verbindlich oder werden als kritisch fuer den Erfolg des Projekts angesehen, wohingegen jene ohne Sternchen optional sind.

Es wird empfohlen, dass das Scrum Team und alle Personen, die in die Scrum Strukturen und Prozesse eingeführt werden, ihren Fokus vorwiegend auf die verbindlichen Inputs, Hilfsmittel und Outputs richten. Product Owner, Scrum Master und weitere fortgeschrittene Scrum Praktiker werden ein genaueres Verständnis des gesamten Kapitels erreichen wollen. Es ist außerdem wichtig, sich bewusst zu machen, dass obwohl alle Prozesse einzeln im SBOK® definiert sind, sie nicht unbedingt sequentiell oder getrennt angewendet werden. Manchmal empfiehlt es sich, je nach spezifischen Bedürfnissen eines jeden Projektes, einige Prozesse miteinander zu kombinieren.

Dieses Kapitel ist aus der Perspektive eines einzelnen Scrum Teams geschrieben worden, welches an einem Sprint arbeitet, um potenziell versandfähige Ergebnisse als Teil eines größeren Projektes, eines Programms oder eines Portfolios zu produzieren. Weitere Informationen, die sich auf den Gebrauch von Scrum für große Projekte beziehen, sind in Kapitel 13 zu finden, die Skalierung von Scrum für Programme und Portfolios befinden sich in Kapitel 14.

Nachdem die Initiierungsphase abgeschlossen ist, können die iterativen Sprint-Zyklen beginnen. Planung und Abschätzung ist die erste von drei Phasen, die sich in jedem Sprint-Zyklus wiederholen. Zu Beginn eines Sprints planen der Product Owner und das Scrum Team, unterstützt durch den Scrum Master, den Sprint. Der Product Owner verfeinert die Epics mit der höchsten Priorität in eine Reihe von gut geschriebenen und abgeschätzten User Storys, zu deren Fertigstellung sich das Scrum Team im kommenden Sprint auf der Grundlage von Annahmen zur Team-Velocity verpflichtet. Der Scrum Master und das Scrum Team erstellen und aktualisieren das Sprint Backlog mit der Liste der User Storys, die das Team für den Sprint zur Fertigstellung zugesagt hat.

Das Scrum Team plant dann seine Arbeit im Detail, indem es die Aufgaben identifiziert und optional abschätzt, die es erledigen muss, um die User Storys für den Sprint zu liefern. Als letzter Planungsschritt für den Sprint vervollständigt das Team das Sprint Backlog mit den Details der Aufgaben und, falls vorhanden, deren Abschätzungen. Das Sprint Backlog wird in der Implementierungsphase verwendet, um den Fortschritt des Teams während des Sprints zu verfolgen. Es ist auch wichtig zu wissen, dass, obwohl alle Phasen und Prozesse im SBOK® Guide eindeutig definiert sind, sie nicht unbedingt sequenziell oder separat durchgeführt werden. Je nach den spezifischen Anforderungen des jeweiligen Projekts kann es manchmal sinnvoller sein, dass sich einige Phasen und/oder Prozesse überlappen.

Abbildung 9-1 gibt einen Überblick über die Prozesse der Planungs- und Abschätzungsphase, welche nachstehend gelistet sind:

9.1 User Storys erstellen — In diesem Prozess werden User Storys und die zugehörigen Abnahmekriterien vom Product Owner erstellt (ausgehend von den zuvor definierten Epics) und in das priorisierte Produkt Backlog aufgenommen. User Storys sollen sicherstellen, dass die Anforderungen des Kunden klar dargestellt werden und von den Business Stakeholdern vollständig verstanden werden können. User Storys müssen greifbar genug sein und die Definition von Bereit erfüllen, bevor sie vom Scrum Team abgeschätzt und entwickelt werden können.

9.2 User Storys abschätzen — In diesem Prozess schätzt das Scrum Team, unterstützt vom Scrum Master, die User Storys ab und ermittelt den Aufwand für die Entwicklung der in jeder User Story beschriebenen Funktionalität. Nur User Storys, die der Definition von Bereit entsprechen und vom Product Owner angemessen definiert sind, werden vom Team abgeschätzt.

9.3 User Storys zusagen — In diesem Prozess verpflichtet sich das Scrum Team, eine Reihe von User Storys für den Sprint zu liefern. Die Entscheidung darüber, welche User Storys zugesagt werden, basiert auf der relativen wertbasierten Priorität der User Storys und dem geschätzten Aufwand und der Team-Velocity für einen Sprint. Im Rahmen dieses Prozesses beginnt das Scrum Team mit der Erstellung des Sprint Backlogs, das die zugesagten User Storys für einen spezifischen Sprint enthält. Das Backlog wird im weiteren Verlauf der Sprintplanung mit Details auf Aufgabenebene verfeinert. Mit dieser Zusage des Scrum Teams, die zu Beginn eines Sprints im Rahmen der Sprintplanung gegeben wird, ist der Inhalt des Sprints definiert und kann nicht mehr geändert werden, sobald die Umsetzungsphase des Sprints begonnen hat.

9.4 Aufgaben identifizieren — In diesem Prozess werden die zugesagten User Storys in konkrete Aufgaben zerlegt und in einer Aufgabenliste zusammengestellt. Die Identifizierung der Aufgaben kann entweder zu Beginn des Sprints für alle zugesagten User Storys erfolgen oder bevor das Team mit der Arbeit an den für jede User Story erforderlichen Aufgaben beginnt.

9.5 Aufgaben abschätzen — Dies ist ein optionaler Prozess, der die Erstellung von Aufgabenschätzungen beinhaltet, wenn das Scrum Team dies für sinnvoll hält. In diesem Prozess schätzt das Scrum Team den Aufwand ab, der für die Erledigung jeder Aufgabe in der Aufgabenliste erforderlich ist. Aufgabenschätzungen können entweder zu Beginn des Sprints für alle User Storys /Aufgaben, die für diesen Sprint relevant sind, oder für jede Aufgabe kurz bevor das Team mit der Arbeit an der jeweiligen User Story/Aufgabe beginnt, ermittelt werden. Die Schätzung kann mit denselben Methoden durchgeführt werden, die für den Prozess "User Storys schätzen" verwendet wurden.

9.6 Sprint Backlog aktualisieren — In diesem Prozess aktualisiert das Scrum Kernteam das Sprint Backlog mit den Aufgabedetails und, falls verfügbar, den Aufgabenschätzungen. Das aktualisierte Sprint Backlog wird in der Implementierungsphase verwendet, um den Fortschritt des Teams während des laufenden Sprints zu verfolgen.

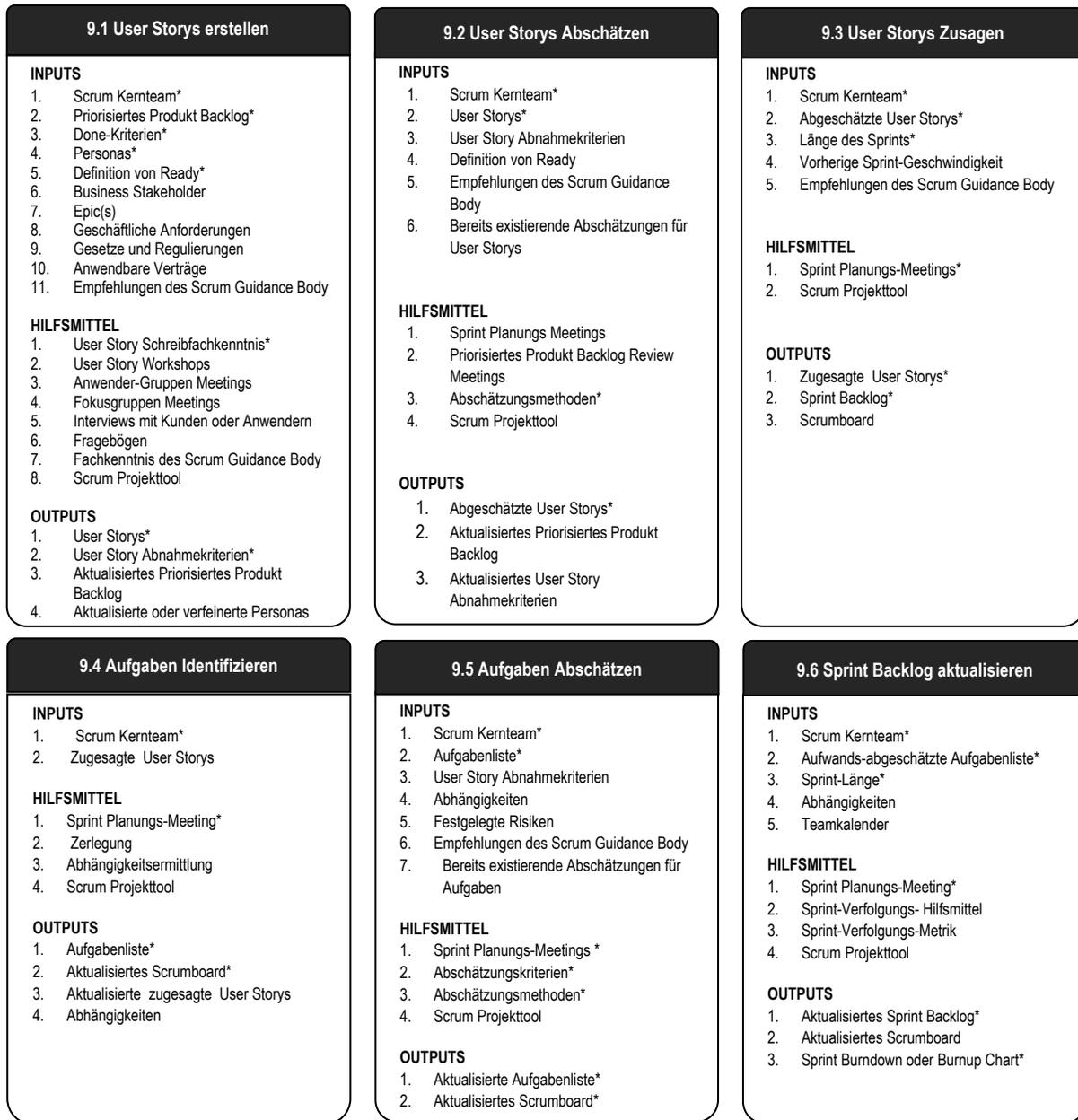


Abbildung 9-1: Planungs- und Abschätzungsübersicht

Abbildung 9-2 unten zeigt die verbindlichen Inputs, Hilfsmittel und Outputs für Prozesse in Planungs- und Schätzungsphase.

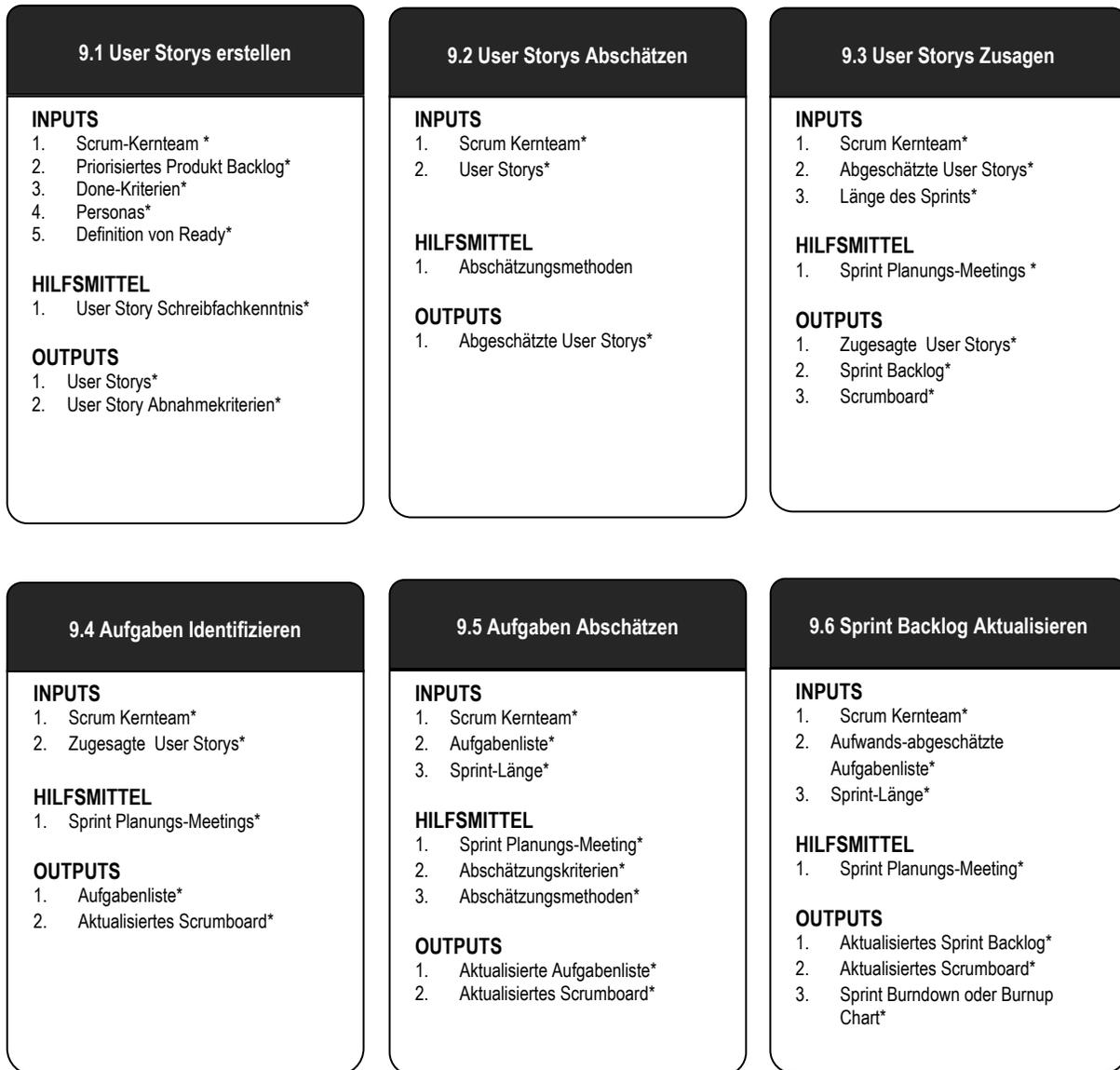


Abbildung 9-2: Planungs- und Abschätzungsübersicht (Kernpunkte)

9.1 User Storys erstellen

In diesem Prozess werden User Storys und die zugehörigen Abnahmekriterien vom Product Owner erstellt (ausgehend von den zuvor definierten Epics) und in das priorisierte Produkt Backlog aufgenommen. User Storys sollen sicherstellen, dass die Anforderungen des Kunden klar dargestellt werden und von den Business Stakeholdern vollständig verstanden werden können. User Storys müssen greifbar genug sein und die Definition von Bereit erfüllen, bevor sie vom Scrum Team abgeschätzt und entwickelt werden können.

User-Story Workshops können genutzt werden, um den Mitgliedern des Scrum-Teams zu helfen, die vom Product Owner erstellten User Storys besser zu verstehen.

Abbildung 9-3 zeigt Inputs, Hilfsmittel und Outputs für den *User Storys Erstellen* Prozess.

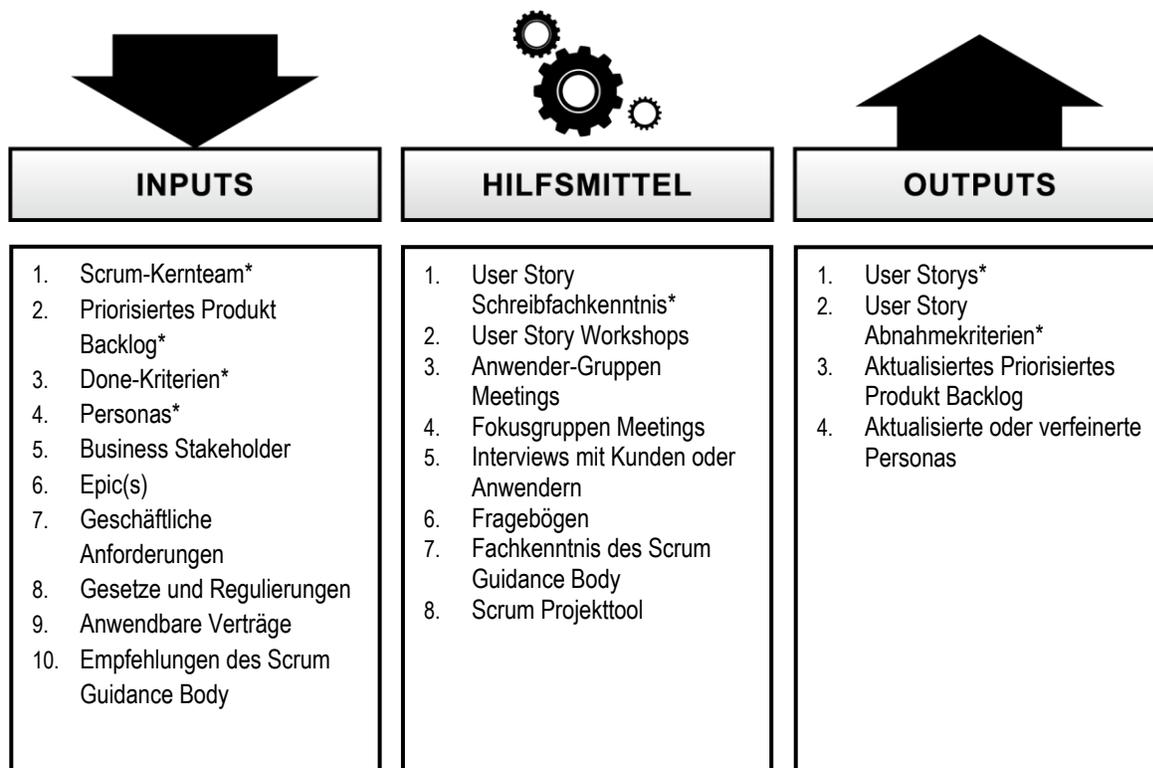


Abbildung 9-3: User Storys erstellen–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

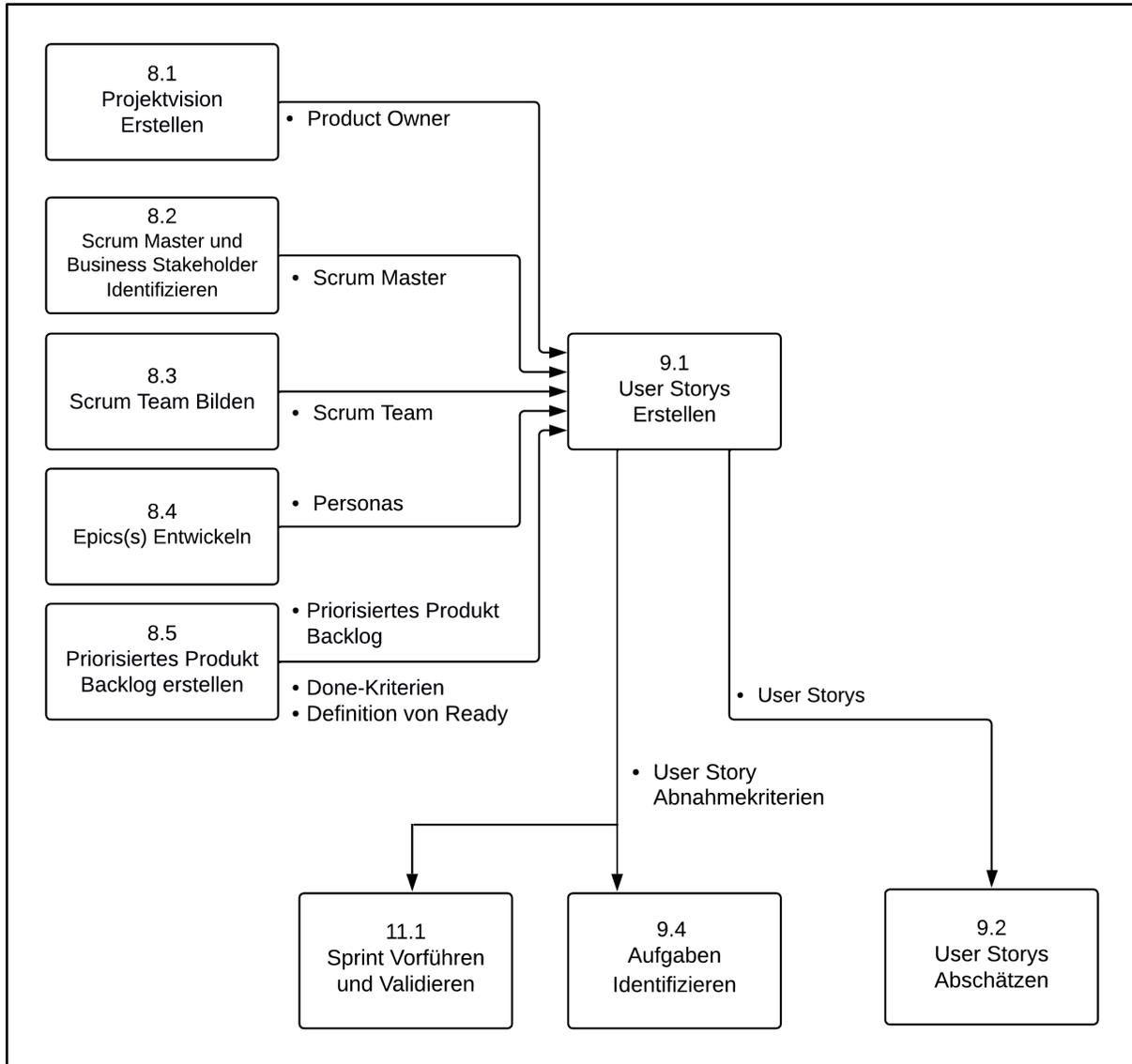


Abbildung 9-4: User Storys erstellen—Datenflussdiagramm

9.1.1 Inputs

9.1.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

9.1.1.2 Priorisiertes Produkt Backlog*

Siehe Abschnitt 8.5.3.1.

9.1.1.3 Done-Kriterien*

Siehe Abschnitt 8.5.3.2.

9.1.1.4 Personas*

Siehe Abschnitt 8.4.3.2.

9.1.1.5 Definition von Bereit (Definition of Ready)*

Die Definition von Bereit definiert die Kriterien, die eine User Story erfüllen muss, bevor sie für die Abschätzung oder Aufnahme in einen Sprint in Frage kommt.

9.1.1.6 Business Stakeholdern

Siehe Abschnitt 8.2.3.2.

9.1.1.7 Epic(s)

Siehe Abschnitt 8.4.3.1.

9.1.1.8 Geschäftliche Anforderungen

Siehe Abschnitt 8.5.1.7.

9.1.1.9 Gesetze und Regulierungen

Siehe Abschnitt 8.4.1.8.

9.1.1.10 Anwendbare Verträge

Siehe Abschnitt 8.4.1.9.

9.1.1.11 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11.

Im Prozess *User Storys Erstellen* können die Empfehlungen des Scrum Guidance Body, Information über Regeln, Standards und die besten Praktiken enthalten, um effektive User Storys zu erstellen.

9.1.2 Hilfsmittel

9.1.2.1 User Story Schreibfachkenntnis*

Basierend auf seiner oder ihrer Interaktion mit den Business Stakeholdern, eigenen Kenntnissen und Know-how des Geschäfts, sowie Inputs aus dem Team, entwickelt der Product Owner User Storys, die das anfängliche Priorisierte Produkt Backlog für das Projekt bilden. Das Priorisierte Produkt Backlog gibt die Gesamtsumme dessen wieder, was im Laufe des Projektes erledigt werden muss. Das Ziel dieser Übung ist es, ausgearbeitete und verfeinerte User Storys hervorzubringen, welche durch das Scrum Team abgeschätzt und umgesetzt werden können. Auch, wenn es hauptsächlich die Aufgabe des Product Owners ist, die User Storys zu schreiben und er dieser Aufgabe auch oft selbst nachgeht, kann bei Bedarf ein User Story Schreib-Workshop durchgeführt werden.

9.1.2.2 User Story Workshops

Siehe Abschnitt 8.4.2.2.

9.1.2.3 Anwender-Gruppen Meetings

Siehe Abschnitt 8.4.2.1.

9.1.2.4 Fokusgruppen Meetings

FokusgruppenMeetings sind eine qualitative Technik, um die Bedürfnisse und Erwartungen der Anwender an ein vorgeschlagenes Produkt abzuschätzen und zu verstehen. Eine kleine Gruppe Anwender wird ausgewählt, um die Fokusgruppe zu bilden. Diese Gruppe kann zufällig aus einer großen Anzahl von Anwendern zusammengestellt oder speziell ausgewählt werden, um alle bedeutenden Personas zu repräsentieren, die zur Zielgruppe gehören. FokusgruppenMeetings folgen normalerweise einem bestimmten Format, bei welchem der Gruppe Fragen gestellt werden, die dann innerhalb der Gruppe diskutiert werden. Jedes FokusgruppenMeeting kann über seine eigenen Diskussionsregeln verfügen, wenn die Organisatoren dies entscheiden. Diese Meetings finden normalerweise in Anwesenheit eines Moderators statt.

9.1.2.5 Interviews mit Kunden oder Anwendern

Siehe Abschnitt 8.4.2.4.

9.1.2.6 Fragebögen

Siehe Abschnitt 8.4.2.5.

9.1.2.7 Fachkenntnis des Scrum Guidance Body

Siehe Abschnitt 8.4.2.7.

Während der Erstellung von User Storys, Scrum Guidance Body und Fachkenntnis des Scrum Guidance Body könnten sich diese auf die dokumentierten Regeln und Regulierungen oder auf die Standards und besten Praktiken beziehen, um User Storys zu erstellen. Es kann ebenfalls ein Team aus Fachexperten existieren, welches dem Product Owner assistieren oder Beratung über User Storys und User Storys erstellen, geben. In diesem Team könnten beispielsweise Wirtschaftsanalysten, führende Architekten, Senior-Entwickler, Scrum Experten oder andere sachkundige Personen sein. Die Expertengruppe besteht normalerweise nicht aus dem gleichen Team, welches bleiben und an diesem Projekt arbeiten wird. Vielmehr tendiert sie dazu, von Projekt zu Projekt zu wechseln und das Scrum Team bei Bedarf zu beraten.

9.1.3 Outputs

9.1.3.1 User Storys*

User Storys halten an einer speziellen, vordefinierten Struktur fest und stellen einen vereinfachten Weg dar, um die Anforderungen und angestrebte End-Anwender Funktionalität zu dokumentieren. Eine User Story verrät drei Dinge über die Anforderungen: wer, was und warum. Die Anforderungen, welche in den User Storys beschrieben werden, sind kurze, einfache und leicht verständliche Aussagen, die eine verstärkte Kommunikation zwischen den Business Stakeholdern und bessere Abschätzungen des Teams zur Folge haben. Einige User Storys können mitunter zu gross sein, um in nur einem Sprint behandelt zu werden. Diese grossen User Storys werden häufig als Epics bezeichnet. Wenn sie im Priorisierten Produkt Backlog zur Erledigung im kommenden Sprint anstehen, werden sie in verschiedene, kleinere User Storys unterteilt. Das Priorisierte Produkt Backlog ist eine dynamische Liste, welche wegen Re-Priorisierung und auf Grund neuer, aktualisierter, verfeinerter und manchmal auch gelöschter User Storys kontinuierlich aktualisiert wird. Die Aktualisierungen des Backlog sind typischerweise das Ergebnis der Änderung der geschäftlichen Anforderungen.

Beachten Sie auch Abschnitt 8.5.3.1, um mehr über das Priorisierte Produkt Backlog zu erfahren.

User Story Format:

Als ein <rolle/persona>, sollte ich in der Lage sein, <Anforderung>, sodass <Nutzen>.

User Story Beispiel:

Als Datenbankadministrator, sollte ich in der Lage sein, eine ausgewählte Anzahl von Datenbankaktualisierungen rueckgaengig zu machen, sodass die gewünschte Version der Datenbank wieder hergestellt wird.

9.1.3.2 User Story Abnahmekriterien*

Jede User Story hat zugeordnete Abnahmekriterien. User Storys sind subjektiv. Die Abnahmekriterien liefern die erforderliche Objektivität, um beim Sprint Review zu entscheiden, welche User Storys akzeptiert oder abgelehnt werden. Abnahmekriterien liefern Klarheit für das Team, was von einer User Story zu erwarten ist. Dadurch wird die Mehrdeutigkeit aus den Anforderungen entfernt und dies hilft bei der Ausrichtung der Erwartungen. Der Product Owner definiert und kommuniziert die Abnahmekriterien ans Scrum Team. In den Sprint Review Meetings stellen die Abnahmekriterien für den Product Owner den Zusammenhang her, um zu entscheiden ob eine User Story zufriedenstellend ausgeführt wurde. Es liegt in der Verantwortung des Scrum Masters, sicherzustellen, dass der Product Owner die Abnahmekriterien für eine zugesagte Anwender User Story während eines Sprint nicht ändert.

9.1.3.3 Aktualisiertes Priorisiertes Produkt Backlog

Das Priorisierte Produkt Backlog, welches im Prozess *Priorisiertes Produkt Backlog erstellen*, mit den Informationen aus User Storys, Epic(s), Schätzungen für User Storys und User Story Abnahmekriterien erstellt und aktualisiert wird.

Das Priorisiertes Produkt Backlog wird in Abschnitt 8.5.3.1. beschrieben.

9.1.3.4 Aktualisierte oder verfeinerte Personas

Personas werden anfangs während des Prozesses *Epic(s) entwickeln* erstellt. Während des Schreibens der User Storys kann es vorkommen, dass das Scrum Team zu der gemeinschaftlichen Entscheidung kommt, dass einig dieser anfänglichen Personas inadäquat sind und deswegen verfeinert werden müssen. Wenn eine Verfeinerung der Personas von Nöten ist, geschieht dies normalerweise nahe dem Ende des Prozesses *User Storys erstellen*.

Personas werden in Abschnitt 8.4.3.2. beschrieben.

9.2 User Storys Abschätzen

In diesem Prozess schätzt das Scrum Team, unterstützt vom Scrum Master, die User Storys ab und ermittelt den Aufwand für die Entwicklung der in jeder User Story beschriebenen Funktionalität. Nur User Storys, die der Definition von Bereit entsprechen und vom Product Owner angemessen definiert sind, werden vom Team abgeschätzt.

Der Product Owner spielt bei der Abschätzung der User Storys keine aktive Rolle, kann aber zur Klärung von Fragen, die das Scrum Team im Zusammenhang mit den abzuschätzenden User Storys hat, hinzugezogen werden.

Abbildung 9-5 zeigt um so Inputs, Hilfsmittel und Outputs für *User Storys Abschätzen* Prozess.

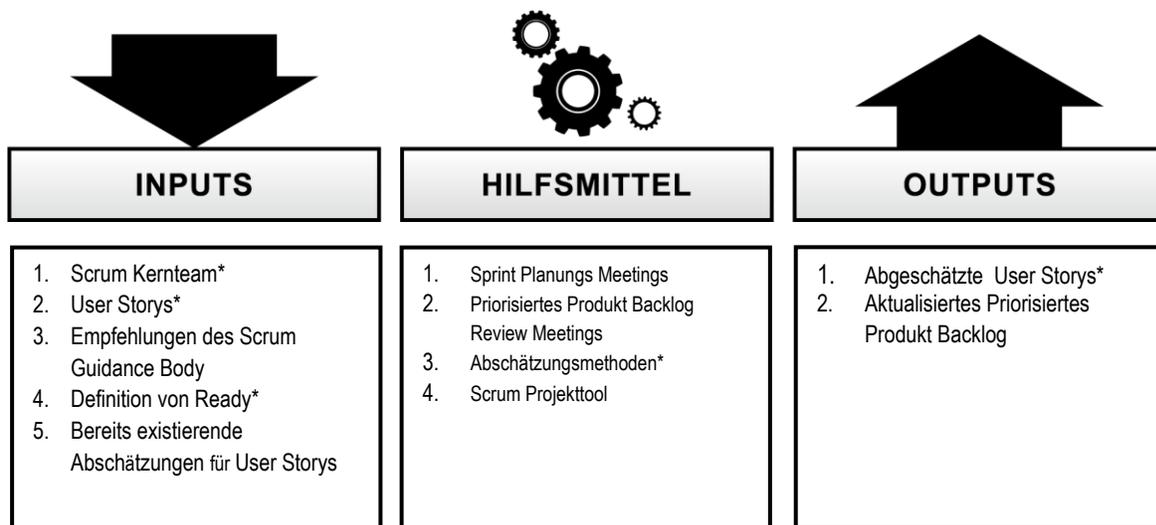


Abbildung 9-5: User Storys Genehmigen, Abschätzen und Zusage-Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

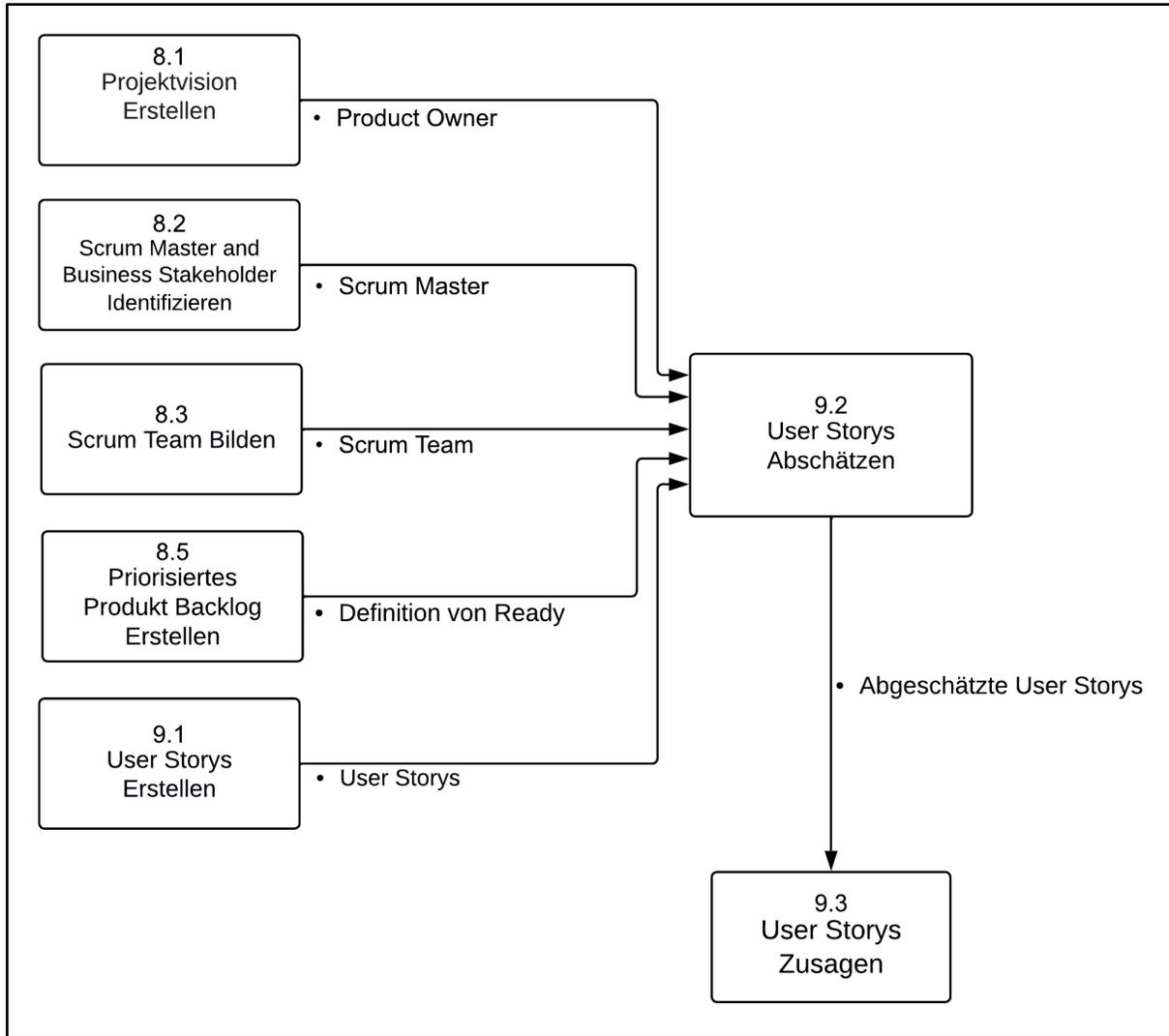


Abbildung 9-6: User Storys Abschätzen–Datenflussdiagramm

9.2.1 Inputs

9.2.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

9.2.1.2 User Storys*

Siehe Abschnitt 9.1.3.1.

9.2.1.3 Definition von Bereit (Definition of Ready)*

Die Definition von Bereit definiert die Kriterien, die eine User Story erfüllen muss, bevor sie für die Abschätzung oder Aufnahme in einen Sprint in Frage kommt.

9.2.1.4 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11.

Im Prozess *User Storys Abschätzen zusagen* können die Empfehlungen des Scrum Guidance Body Informationen über Regeln, Regularien, Standards enthalten sowie zu den besten Verfahren, die benötigt werden, um User Storys effektiv zu abzuschätzen.

9.2.1.5 Bereits vorhandene Abschätzungen von User Storys*

Möglicherweise gibt es bereits Abschätzungen des Aufwands für die Fertigstellung von User Storys, insbesondere dann, wenn ähnliche User Storys bereits in demselben Projekt oder in verwandten früheren Projekten fertiggestellt wurden. Aus dem Aufwand und der Zeit, die für die Fertigstellung ähnlicher User Storys benötigt wurden, lassen sich Schätzungen für den Zeitaufwand zur Fertigstellung vorhandener User Storys ableiten. Experten, die in der Vergangenheit ähnliche Projektanforderungen umgesetzt haben, können möglicherweise auch einige Aufwandsabschätzungen für die User Storys liefern. Diese bereits existierenden Abschätzungen helfen dem Scrum Team, die User Storys für den Sprint gründlich zu schätzen und zuzusagen. Es ist wichtig sicherzustellen, dass das Scrum Team seine eigenen Abschätzungen der User Storys erstellt, bevor die User Storys als Teil des Sprints zugesagt werden, und sich nicht ausschließlich auf bereits vorhandene Abschätzungen für User Storys verlässt.

9.2.2 Hilfsmittel

9.2.2.1 Abschätzungsmethoden

Wenn neue oder aktualisierte User Storys im Product Backlog verfeinert werden, wird für jede User Story vom Scrum Team entweder eine neue Abschätzung erstellt oder eine bereits existierende aktualisiert. Relatives Sizing oder Story Points können für die Schätzung der Gesamtgröße einer User Story oder eines Features verwendet werden. Bei diesem Ansatz wird ein Story Point-Wert auf der Grundlage einer Gesamtbewertung der Größe einer User Story unter Berücksichtigung des damit verbundenen Risikos, des erforderlichen Aufwands und des Komplexitätsgrads zugewiesen. Diese Bewertung wird vom Scrum-Team durchgeführt, und es wird ein Story-Point-Wert zugewiesen. Sobald die Bewertung einer User Story aus dem priorisierten Product Backlog abgeschlossen ist, kann das Scrum Team die anderen User Story im Verhältnis zu dieser ersten Story bewerten. Es ist zu beachten, dass die Kalibrierung der Story-Punkte für jedes Team unterschiedlich ist, so dass die Anzahl der fertiggestellten Story Points nicht für einen Vergleich zwischen den Teams verwendet werden kann. Darüber hinaus hängt die gewählte Schätzmethode davon ab, wie detailliert die Schätzung für das Team sein muss.

Einige Methoden, die zur Abschätzung von User Storys verwendet werden können, sind:

1. Breitband Delphi Technik

Breitband-Delphi ist eine gruppenbasierte Technik zur Aufwandsabschätzung. Jedes einzelne Teammitglied gibt anonym Schätzungen für jedes Feature ab. Diese ersten Schätzungen werden dann in einem Diagramm dargestellt. Das Team erörtert dann die Faktoren, welche die jeweiligen Schätzungen beeinflusst haben und geht dann in eine zweite Runde der Abschätzungen. Dieser Vorgang wird wiederholt, bis die individuellen Schätzungen nahe beieinanderliegen und ein Konsens für die Abschätzung erreicht werden kann.

2. Planungspoker

Planungspoker, auch Schätzungspoker genannt, ist ein Verfahren zur Abschätzung, welches auf Konsens basiert, um die relative Größe der User Storys oder den benötigten Aufwand abzuschätzen.

Im Planungspoker wird jedem Teammitglied ein Kartendeck zugewiesen, wobei jede Karte durchgehend nummeriert ist. Die Nummern stehen für die Komplexität des Problems, bezogen auf Zeit oder Aufwand, welche durch die Teammitglieder geschätzt worden sind. Die Mitglieder des Scrum Teams bewerten die User Story und versuchen sie besser zu verstehen, bevor sie ihre Schätzung für ihre Entwicklung abgeben. Dann nimmt jedes Mitglied eine Karte vom Deck, welche die Schätzung für die User Story repräsentiert. Wenn die Mehrheit oder alle Teammitglieder die gleiche Karte wählen, dann wird die Schätzung, welche durch diese Karte dargestellt wird, angewendet. Wenn es keine Einigung gibt, dann müssen die Teammitglieder über die Gründe der Wahl unterschiedlicher Karten oder Abgabe von Schätzungen diskutieren. Nach der Diskussion werden die Karten erneut gezogen. Diese Reihenfolge wird so lange fortgeführt, bis alle Voraussetzungen verstanden, Missverständnisse geklärt und ein Konsens oder eine Übereinstimmung getroffen worden ist. Planungspoker bietet eine größere Interaktion und eine verbesserte Kommunikation zwischen den Teilnehmern. Es erleichtert unabhängiges Denken durch die Teilnehmer während das Phänomen des Gruppendenkens vermieden wird.

3. Faust von Fünf

Unter Faust von Fünf versteht man einen einfachen und schnellen Mechanismus, der sowohl als eine Praktik zur Abschätzung als auch als Technik zur Konsensbildung innerhalb einer Gruppe verwendet werden kann. Nach einer anfänglichen Diskussion über einen bestimmten Abschätzungsgegenstand werden die einzelnen Scrum Teammitglieder aufgefordert, mithilfe ihrer Finger auf einer Skala von 1 bis 5 abzustimmen. Wird Faust von Fünf zur Abschätzung verwendet gibt die Anzahl der gezeigten Finger den geschätzten relativen Wert an. Teammitglieder mit den höchsten und niedrigsten Werten erklären dem Team ihre Begründung für ihre Abschätzung und dies wird dann diskutiert. Nachdem das Team sein Diskussion beendet hat, findet entweder eine weitere Runde der Faust von Fünf statt oder es gibt eine Teamentscheidung.

Der Wert der Anwendung dieser Technik liegt nicht nur darin, einen Konsens herzustellen, sondern ebenfalls eine Diskussion anzuregen, da jedes Teammitglied dazu aufgefordert wird, die Gründe für die eigene Abschätzung zu erklären. Es wird ihnen ebenfalls ermöglicht, alle Probleme oder Bedenken anzusprechen. Wird Faust von Fünf als Technik zur Konsensbildung verwendet, wird der Vorschlag oder die anstehende Entscheidung zunächst diskutiert. Danach stimmen die Teammitglieder entsprechend dem Grad ihrer Zustimmung und ihrem Wunsch nach weiterer Diskussion ab.

Ein Finger: Ich stimme dem Gruppenergebnis nicht zu und habe große Bedenken.

Zwei Finger: Ich stimme dem Gruppenergebnis nicht zu und würde gerne einige kleinere Probleme diskutieren.

Drei Finger: Ich bin mir nicht sicher und würde mich gerne dem Gruppenergebnis, dem Konsens anschließen.

Vier Finger: Ich stimme dem Gruppenergebnis zu und würde gerne einige kleinere Probleme diskutieren.

Fünf Finger: Ich stimme dem Gruppenergebnis von ganzem Herzen zu.

4. Vergleichende Abschätzung

Vergleichende Abschätzung (Affinity Estimation), wie z. B. T-Shirt-Größen, ist eine Technik zur schnellen Abschätzung einer großen Anzahl von User Storys. Die User Storys werden dann von den Teammitgliedern mit Post-its oder Indexkarten und Klebeband an einer Wand oder auf einer anderen Oberfläche, in der Reihenfolge von klein zu groß, befestigt. Dafür beginnt jedes Teammitglied mit einer Teilmenge der User Storys aus dem generellen Priorisierten Produkt Backlog, die nach relativer Größe angeordnet werden. Diese Anfängliche Anordnung wird in Stille gemacht. Wenn alle ihre User Storys an der Wand platziert haben, beurteilt das Team die Anordnung und verschiebt User Storys angemessen. Der zweite Teil dieser Übung beinhaltet die Diskussion. Schließlich wird der Product Owner einige Größenkategorien an der Wand zeigen. Diese Kategorien können klein, mittel oder groß oder nummeriert sein, um die Story-Punktwerte und die relative Größe anzuzeigen. Das Team wird dann die User Storys in diese Kategorien verschieben, was den letzten Schritt dieses Prozesses darstellt. Der Schlüsselvorteil dieses Ansatzes ist ein sehr transparenter Prozess, der für jeden sichtbar und leicht durchzuführen ist.

9.2.2.2 Sprint Planungs Meetings

Während Sprint Planungs-Meetings werden User Storys vom Scrum Team besprochen. Falls das nicht bereits während der Erstellung oder der Pflege des Product Backlogs erfolgt ist, wird jede User Story bewertet und mit einer high-level Aufwandsabschätzung versehen.

Siehe auch Abschnitte 9.3.2.1, 9.4.2.1 und 9.6.2.1.

9.2.2.3 Priorisiertes Produkt Backlog Review Meetings

Priorisiertes Produkt Backlog Review Meetings werden als Teil des *Priorisiertes Produkt Backlog verfeinern* Prozesses durchgeführt. Die Informationen aus diesen Treffen bringen zusätzliche Klarheit über die User Storys und helfen bei der Festlegung ihrer Abschätzung.

9.2.3 Outputs

9.2.3.1 Abgeschätzte User Storys*

Nachdem die User Storys vom Scrum Team unter Anwendung verschiedener Schätzungstechniken, welche in diesem Abschnitt beschrieben werden, abgeschätzt wurden, sind sie abgeschätzte User Storys.

9.2.3.2 Aktualisiertes Priorisiertes Produkt Backlog

Siehe Abschnitt 9.2.3.2.

9.3 User Storys Zusagen

In diesem Prozess verpflichtet sich das Scrum Team, eine Reihe von User Storys für den Sprint zu liefern. Die Entscheidung darüber, welche User Storys zugesagt werden, basiert auf der relativen wertbasierten Priorität der User Storys und dem geschätzten Aufwand und der Team-Velocity für einen Sprint. Im Rahmen dieses Prozesses beginnt das Scrum Team mit der Erstellung des Sprint Backlogs, das die zugesagten User Storys für einen spezifischen Sprint enthält. Das Backlog wird im weiteren Verlauf der Sprintplanung mit Details auf Aufgabenebene verfeinert.

Mit dieser Zusage des Scrum Teams, die zu Beginn eines Sprints im Rahmen der Sprintplanung gegeben wird, ist der Inhalt des Sprints definiert und kann nicht mehr geändert werden, sobald die Umsetzungsphase des Sprints begonnen hat.

Abbildung 9-7 zeigt Inputs, Hilfsmittel und Outputs für den *User Storys Zusagen* Prozess.

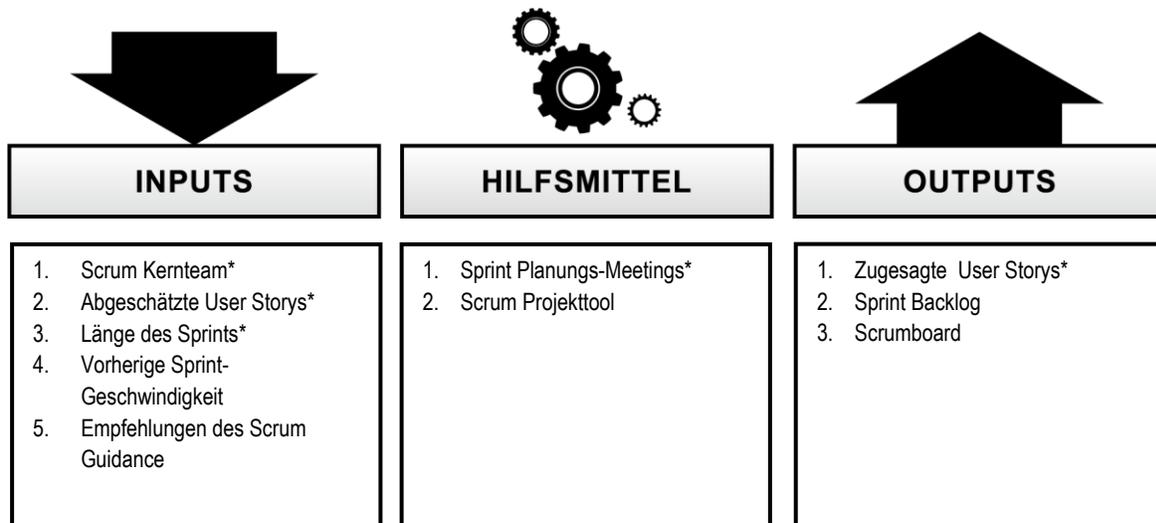


Abbildung 9-7: User Storys Zusagen–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

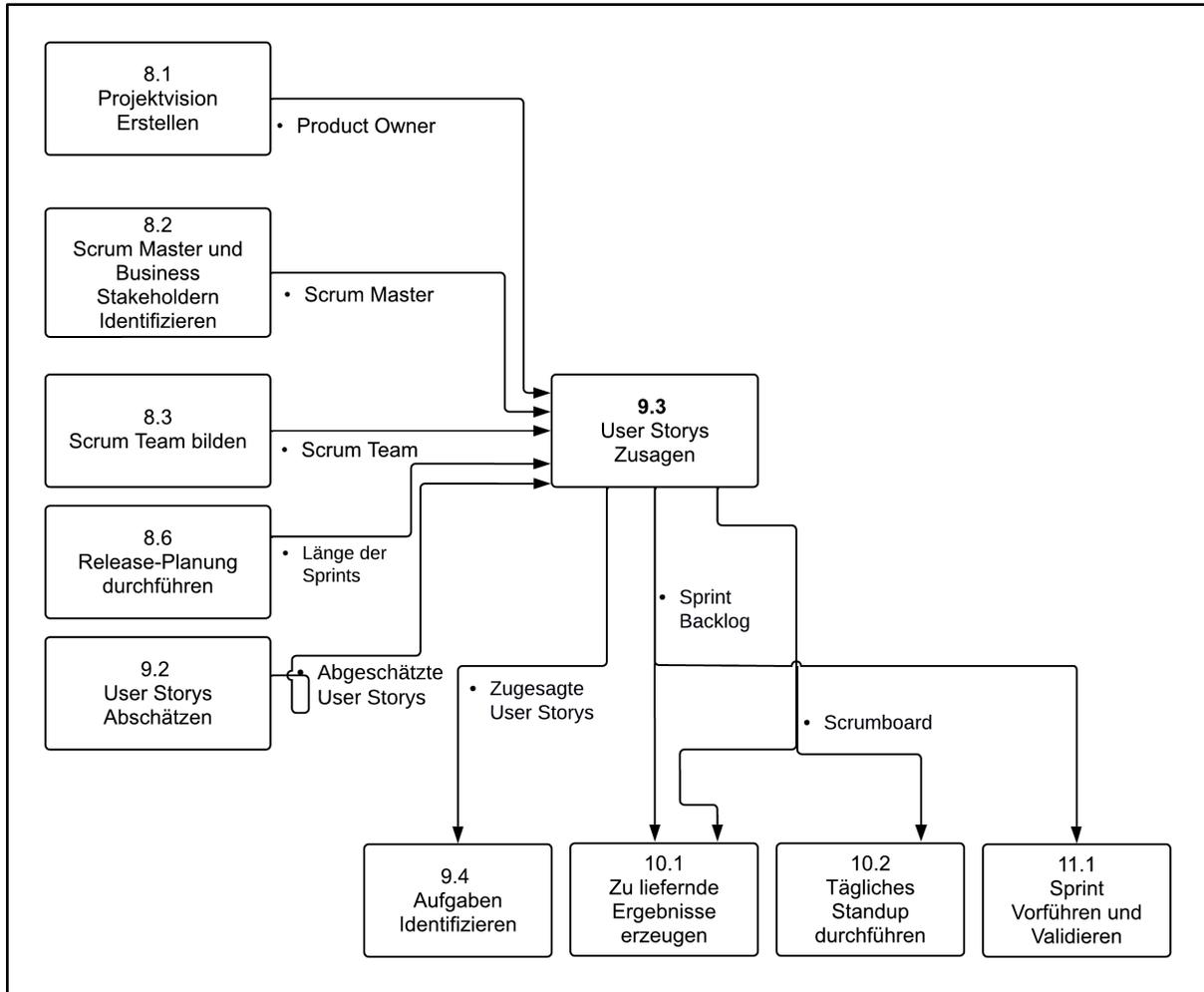


Abbildung 9-8: User Storys Zusagen–Datenflussdiagramm

9.3.1 Inputs

9.3.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

9.3.1.2 Abgeschätzte User Storys*

Siehe Abschnitt 9.2.3.1.

9.3.1.3 Länge des Sprints*

Siehe Abschnitt 8.6.3.2.

9.3.1.4 Vorherige Sprint-Geschwindigkeit

Sprint-Geschwindigkeit ist die Geschwindigkeit, in der das Team die Arbeit in einem Sprint erledigen kann. Sie wird in der Regel in den gleichen Einheiten ausgedrückt, wie sie beim Schätzen verwendet werden, normalerweise Story Points oder Idealzeit. Es wird ein Verzeichnis der Sprint-Geschwindigkeit des Teams für den Sprint geführt und als Referenz für zukünftige Sprints verwendet. Die vorherige Sprint-Geschwindigkeit wird zum wichtigsten Faktor bei der Festlegung des Arbeitsvolumens, welches das Team in einem anschließenden Sprint zusagen kann. Jegliche Änderungen der Situation oder Bedingungen seit dem letzten Sprint werden erklärt, um die exakte Dauer der Sprint-Geschwindigkeit eines kommenden Sprint sicherzustellen.

9.3.1.5 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11.

Im Prozess *User Storys Abschätzen* können die Empfehlungen des Scrum Guidance Body Informationen über Regeln, Regularien, Standards enthalten sowie zu den besten Verfahren, die benötigt werden, um User Storys effektiv zu abzuschätzen.

9.3.2 Hilfsmittel

9.3.2.1 Sprint Planungs-Meetings*

In den Sprint Planungs-Meetings kommt das Scrum Team zusammen, um die Arbeit, welche im Sprint erledigt werden soll, zu planen. Das Team bewertet die zugesagten User Storys an der Spitze des Priorisierten Produkt Backlogs. Der Product Owner ist bei diesen Meetings anwesend, für den Fall, dass Klarstellungen in Bezug auf die User Story und im Priorisierten Produkt Backlog benötigt werden und, um dem Team zu helfen Designentscheidungen zu treffen. Um sicherzustellen, dass die Gruppe beim Thema bleibt, sollten solche Meetings einen Zeitrahmen haben, mit einer Standarddauer bis zu zwei Stunden pro Woche an Sprint-Dauer. Dies hilft der Tendenz vorzubeugen, in Diskussionen abzudriften, welche eigentlich Bestandteil anderer Meetings sein sollten, wie beispielsweise das Release Planungs- oder das Sprint Review Meeting. Am Ende des Meetings verpflichtet sich das komplette Scrum Team, eine Untermenge der Storys des Priorisierten Produkt Backlogs im Sprint umzusetzen.

Siehe auch Abschnitte 9.2.2.1, 9.4.2.1, 9.5.2.1 and 9.6.2.1.

9.3.2.2 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

9.3.3 Outputs

9.3.3.1 Zugesagte User Storys*

Das Scrum Team sagt eine Teilmenge der User Storys zu, von denen es glaubt, dass es sie, basierend auf der Team Geschwindigkeit, im nächsten Sprint implementieren kann. Die zugesagten User Storys werden stets gemäß den vom Product Owner definierten Prioritäten ausgewählt.

9.3.3.2 Sprint Backlog*

Die Liste der User Storys, die vom Scrum Team im kommenden Sprint realisiert werden sollen, wird als Sprint Backlog bezeichnet. Dabei handelt es sich um eine Teilmenge des priorisierten Produkt Backlogs, das die festgelegten User Storys enthält, die einem bestimmten Sprint zugewiesen sind. Das Sprint Backlog wird im weiteren Verlauf der Sprintplanung mit Details auf Aufgabenebene verfeinert.

Es ist üblich, dass die User Storys des Sprint Backlogs (und die zugehörigen Aufgaben) auf einem Scrumboard oder einem ähnlichen Task Board dargestellt werden, das den aktuellen Status der User Storys des Produkt Backlogs ständig sichtbar macht. Sobald das Sprint Backlog fertiggestellt und vom Scrum Team bestätigt ist, sollten keine neuen User Storys mehr hinzugefügt werden. Wenn sich während eines Sprints neue Anforderungen ergeben, sollten diese in das priorisierte Product Backlog aufgenommen werden, um für einen zukünftigen Sprint berücksichtigt zu werden.

9.3.3.3 Scrumboard*

Es ist wichtig, den Fortschritt eines Sprints zu verfolgen und zu wissen, wo das Scrum-Team in Bezug auf die Fertigstellung der User Storys (und Aufgaben) im Sprint Backlog steht. Um die Arbeit in einem Sprint zu verfolgen, kann eine Vielzahl von Hilfsmitteln verwendet werden, aber eines der gebräuchlichsten ist ein Scrumboard, auch bekannt als Taskboard oder Fortschrittsdiagramm. Die Transparenz von Scrum ergibt sich aus offen einsehbaren Informationswerkzeugen wie dem Scrumboard, das den laufenden Fortschritt des Teams anzeigt. Das Team verwendet ein Scrumboard, um den Fortschritt während jedes Sprints zu planen und zu verfolgen.

In der einfachsten Version eines Scrumboards ist das Board in drei Abschnitte unterteilt: "Nicht begonnene Arbeiten" (auch als "Zu erledigen" bezeichnet), "Laufende Arbeiten" (auch als "In Arbeit" bezeichnet) und "Abgeschlossene Arbeiten" (auch als "Fertig" bezeichnet). Haftnotizen, die jede User Story und die zugehörigen Aufgaben repräsentieren, werden in der entsprechenden Kategorie platziert, um den Status der im aktuellen Sprint abzuschließenden Arbeit wiederzugeben. Die Aufgabenzettel werden mit dem Fortschreiten der Arbeit in die nächste Kategorie verschoben.

Ein typisches Scrumboard ist in Abbildung 9-9 dargestellt. Das Scrumboard zeigt alle User Storys in der linken Spalte und hat drei Spalten mit den Bezeichnungen "Zu erledigen", "In Arbeit" und "Erledigt". Wenn Aufgaben, die mit den User Storys verbunden sind, identifiziert und in nachfolgenden Prozessen bearbeitet werden, werden diese Aufgaben in den jeweiligen Spalten dargestellt.

User Stories	Aufgaben		
	Zu Erledigen	In Arbeit	Erledigt
1			
2			
3			
4			

Abbildung 9-9: Typisches Scrumboard

Variationen des typischen Scrumboards können verwendet werden, um den Status der Arbeit des Teams genauer darzustellen. Eine Variante enthält zum Beispiel eine zusätzliche Spalte "Testen", um anzuzeigen, dass die Aufgabe bis auf das Testen abgeschlossen ist, und das Arbeitsergebnis gerade getestet wird. Die Spalte "Erledigt" wird in diesem Fall verwendet, um diejenigen Aufgaben darzustellen, die vollständig abgeschlossen wurden, einschließlich der erfolgreichen Tests. Anstelle der Spalte "Testen" kann das Team auch jede andere Spalte in das Scrumboard einfügen, die es für hilfreich hält, um den Fortschritt zu verfolgen. Ein Beispiel für ein Scrumboard mit vier Spalten ist in Abbildung 9-10 dargestellt.

User Stories	Aufgaben			
	Zu Erledigen	In Arbeit	Im Test	Erledigt
1				
2				
3				
4				

Abbildung 9-10: Scrumboard mit vier Abschnitten

Das Scrumboard kann manuell auf Papier oder auf einem großen Whiteboard geführt werden, oder es kann elektronisch in einer Tabellenkalkulation oder mit einem Scrum Projekttool geführt werden. Das Scrum-Team sollte das Scrumboard nach Bedarf anpassen oder ergänzen, so dass das Scrumboard weiterhin genaue visuelle Informationen und Kontrolle hinsichtlich des Status der geleisteten Arbeit bietet (wie vom Team vereinbart und zugesagt).

9.4 Aufgaben Identifizieren

In diesem Prozess werden die zugesagten User Storys in konkrete Aufgaben zerlegt und in einer Aufgabenliste zusammengestellt. Die Identifizierung der Aufgaben kann entweder zu Beginn des Sprints für alle zugesagten User Storys erfolgen oder bevor das Team mit der Arbeit an den für jede User Story erforderlichen Aufgaben beginnt.

Der Product Owner spielt keine aktive Rolle bei der Identifizierung von Aufgaben, sollte aber zur Verfügung stehen, um alle Fragen des Scrum-Teams zu beantworten, die bei der Zerlegung der User Storys in Aufgaben auftreten können.

Abbildung 9-11 zeigt Inputs, Hilfsmittel und Outputs für den *Aufgaben Identifizieren* Prozess.

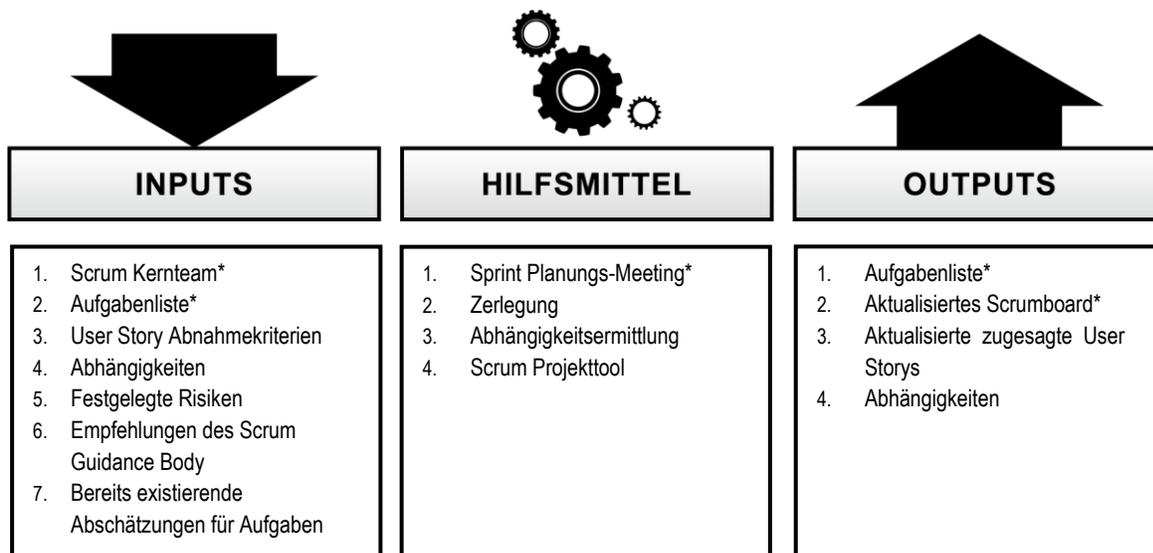


Abbildung 9-11: Aufgaben abschätzen–Inputs,Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

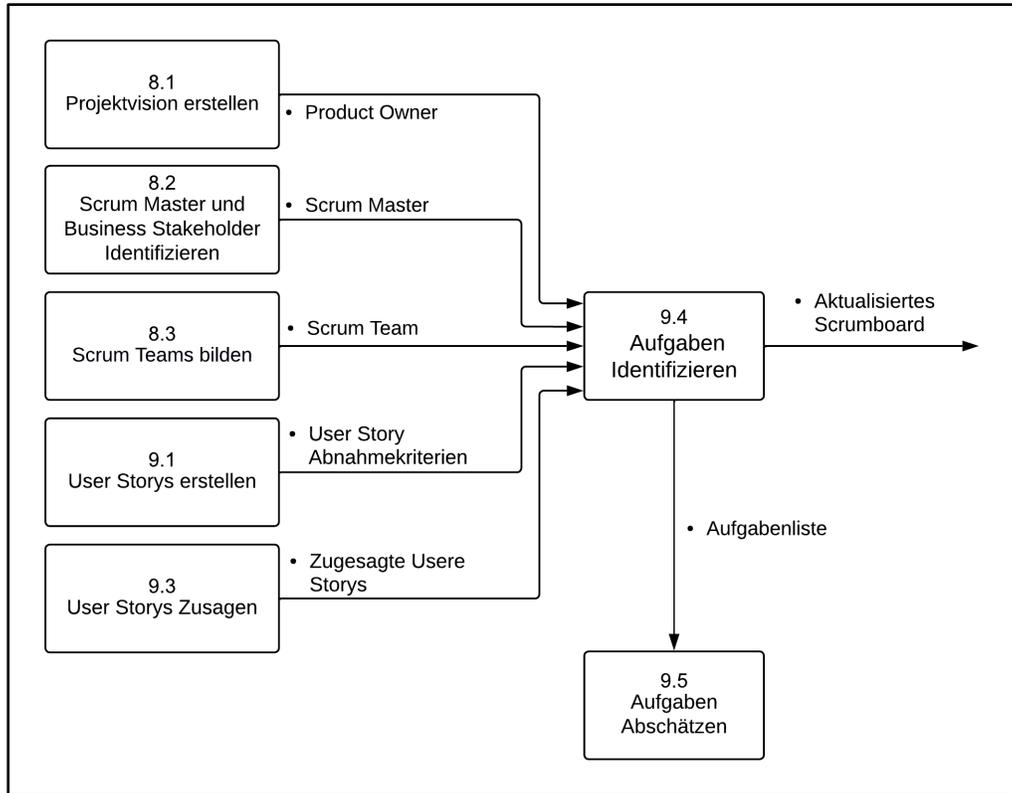


Abbildung 9-12: Aufgabenschätzung-Datenflussdiagramm

9.4.1 Inputs

9.4.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

9.4.1.2 Zugesagte User Storys*

Siehe Abschnitt 9.3.3.1.

9.4.1.3 User Story Abnahmekriterien*

Siehe Abschnitt 9.1.3.2.

Der Product Owner muss sicherstellen, dass die definierten Abnahmekriterien für die User Storys geeignet sind, und er muss auch dem Scrum Team Klarheit über die Anforderungen verschaffen. Das Verständnis der Abnahmekriterien durch das Scrum Team hilft diesem, zu bestimmen, welche Aufgaben erforderlich sind, um die Anforderungen der User Story zu erfüllen.

Das Abnahmetesten bezieht sich auf die Bewertung der Fähigkeit des fertiggestellten Produkts, die Abnahmekriterien zu erfüllen. Dadurch erhält der Product Owner Informationen, die ihm helfen, eine Entscheidung über die Genehmigung oder Ablehnung der zu Liefernden Ergebnisse zu treffen.

Die Abnahmekriterien sollten klar, unmissverständlich und spezifisch sein. Sie sollten so definiert werden, dass das Team in der Lage ist, zu überprüfen, ob die Ergebnisse mit den Zielen des Sponsors übereinstimmen.

9.4.2 Hilfsmittel

9.4.2.1 Sprint Abschätzungs-Meetings*

In den Sprint planungs-Meetings kommt das Scrum Team zusammen, um die Arbeit, welche im Sprint erledigt werden soll, zu planen. Das Team bewertet alle für den zugesagten User Storys und ermittelt / identifiziert alle Aufgaben, die durchgeführt werden müssen, um die zu den User Storys gehörenden zu liefernden Ergebnisse zu erzeugen und die Abnahmekriterien zu erfüllen. Der Product Owner ist bei diesen Meetings anwesend, für den Fall, dass Klarstellungen in Bezug auf die zugesagten User Storys benötigt werden, um dem Team zu helfen Designentscheidungen zu treffen.

Siehe auch Abschnitte 9.2.2.1, 9.3.2.1, 9.5.2.1 and 9.6.2.1.

9.4.2.2 Zerlegung

Die Zerlegung wird vom Scrum-Team verwendet, um die User Storys im kommenden Sprint in detaillierte Aufgaben aufzuspalten. Die User Storys müssen so weit zerlegt werden, dass dem Scrum-Team angemessene Informationen zur Verfügung stehen, um anhand der in der Aufgabenliste erwähnten Aufgaben die zu liefernden Ergebnisse zu erstellen.

9.4.2.3 Abhängigkeitsermittlung

Sobald das Scrum-Team User Storys für einen gegebenen Sprint ausgewählt hat, sollte es danach alle Abhängigkeiten in Betracht ziehen, sowohl die Verfügbarkeit von Menschen als auch technische Abhängigkeiten. Abhängigkeiten ordnungsgemäß zu dokumentieren, hilft den Scrum-Teams dabei, die relative Reihenfolge zu bestimmen, in der Aufgaben ausgeführt werden sollten, um die zu liefernden Ergebnisse des Sprints zu erzeugen. Abhängigkeiten heben auch die Beziehung und Interaktionen hervor, die zwischen verschiedenen Aufgaben bestehen (sowohl innerhalb eines Scrum Teams, das an einem gegebenen Sprint arbeitet als auch zu anderen Scrum Teams im Projekt).

Es gibt verschiedene Abhängigkeitstypen: zwingend und nicht-zwingend, intern und extern oder verschiedene Kombinationen. So kann zum Beispiel eine Abhängigkeit sowohl zwingend als auch extern sein.

- **Zwingende Abhängigkeiten**—Diese Abhängigkeiten ergeben sich entweder aus der Natur der Arbeit, wie etwa eine physische Einschränkung, oder aus vertraglichen Verpflichtungen oder gesetzlichen Anforderungen. So kann zum Beispiel die Arbeit in der ersten Etage nicht beginnen, solange das Fundament des Gebäudes nicht fertig ist. Zwingende Abhängigkeiten werden gewöhnlich auch als harte Logik bezeichnet.
- **Nicht-zwingende Abhängigkeiten**—Nicht-zwingende Abhängigkeiten sind Abhängigkeiten, die nach Wahl in den Arbeitsfluss platziert werden. Nicht-zwingende Abhängigkeiten werden in der Regel durch das Scrum-Team auf der Grundlage von Erfahrung oder von besten Praktiken in einem bestimmten Feld oder einer bestimmten Domäne festgelegt. Das Team kann sich beispielsweise dazu entscheiden, erst eine Aufgabe zu beenden, bevor an einer Neuen gearbeitet wird, weil es die beste Praxis ist, aber nicht, weil es notwendig ist. So kann sich das Team zum Beispiel dazu entscheiden, die Tür- und Fensterrahmen zu bauen, bevor die komplette Struktur der Wand vorhanden ist.
- **Externe Abhängigkeiten**—Externe Abhängigkeiten sind solche, die im Zusammenhang mit Aufgaben, Aktivitäten oder Produkten stehen, die nicht zum Arbeitsumfang des Scrum-Teams gehören, aber erforderlich sind, um eine Aufgabe im Projekt abzuschließen, oder ein zu lieferndes Ergebnis des Projekts zu erzeugen. Externe Abhängigkeiten sind in der Regel außerhalb der Kontrolle des Scrum-Teams. Wenn das Scrum Team beispielsweise nicht dafür verantwortlich ist, die benötigten Materialien zum Bau der Wände zu besorgen, dann werden diese Aufgaben, welche sich mit der Beschaffung befassen als externe Abhängigkeiten betrachtet.
- **Interne Abhängigkeiten**—Interne Abhängigkeiten sind die Abhängigkeiten zwischen Aufgaben, Produkten oder Aktivitäten, die unter der Kontrolle des Scrum Teams sind und zum durch das Scrum Team auszuführenden Arbeitsumfang gehören. Ein Wand muss erst errichtet werden, bevor mit dem Streichen begonnen werden kann.

Dies ist ein Beispiel einer interneren Abhängigkeit, weil beide Aufgaben Teil des Projektes sind. In einem solchen Fall ist sie auch zwingend, da sie auf einer physikalischen Einschränkung basiert. Es ist unmöglich die Wand zu streichen, bevor sie errichtet ist.

9.4.2.4 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

9.4.3 Outputs

9.4.3.1 Aufgabenliste*

Hierbei handelt es sich um eine umfassende Liste, die alle Aufgaben enthält, zu denen sich das Scrum Team im aktuellen Sprint verpflichtet hat, und deren Beschreibungen. Beschreibungen aller Aufgaben sind in dieser Liste zusammen mit den Schätzungen enthalten, die im Prozess Aufgaben Identifizieren hergeleitet worden sind. Die Aufgabenliste muss alle Bemühungen in Bezug auf Tests und Integration enthalten, sodass das Produktinkrement aus dem Sprint erfolgreich in die Lieferungen des vorherigen Sprints integriert werden kann.

Obwohl Aufgaben häufig auf Aktivitäten beruhen, wird der Level der Granularität, in welchen die Aufgaben zerlegt werden, vom Scrum Team beschlossen.

9.4.3.2 Aktualisiertes Scrumboard*

Mit der Identifizierung von Aufgaben wird das Scrumboard aktualisiert, um die mit jeder User Story verbundenen Aufgaben anzuzeigen. Die Aufgaben werden typischerweise auf Haftnotizen auf einem physischen Scrumboard oder als Einträge unter den entsprechenden User Storys angezeigt, wenn ein elektronisches Scrum Projekttool verwendet wird. Während der Implementierung, wenn das Team Aufgaben hinzufügt, zuweist und aktualisiert, wird das Scrumboard mit den zusätzlichen Aufgaben und dem Status jeder Aufgabe aktualisiert. Wenn das Team die Aufgaben abgeschätzt hat, werden die Aufgabenschätzungen auch auf dem Scrumboard dargestellt.

In dem Beispiel in Abbildung 9-13 zeigt das Scrumboard, dass drei User Storys, User Story 1, 2 und 3 bereits in Aufgaben zerlegt wurden, aber User Story 4 wurde noch nicht in Aufgaben zerlegt. Zu Beginn eines Sprints werden alle Aufgaben für diesen Sprint in die Zu erledigen-Spalte eingeordnet und anschließend entsprechend ihrem Fortschritt nach vorne verschoben. User Story 1 hat z.B. 7 Aufgaben, die alle als "Zu erledigen" kategorisiert sind, was bedeutet, dass das Scrum Team noch nicht mit der Arbeit an einer dieser Aufgaben begonnen hat.

Weitere Informationen zum Scrumboard finden Sie in den Abschnitten 9.3.3.3 und 10.1.1.3.

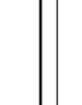
User Storys	Aufgaben		
	Zu Erledigen	In Arbeit	Erledigt
1			
2			
3			
4			

Abbildung 9-13: Scrumboard mit Identifizierten Aufgaben

9.4.3.3 Aktualisierte Zugesagte User Storys

Die User Storys werden während dieses Prozesses aktualisiert. Aktualisierungen können Überarbeitungen der ursprünglichen User Story Schätzungen beinhalten. Diese basieren auf der Aufgabenerstellung und den Komplexitäts-Faktoren, die während des Sprint Planungs-Meetings besprochen wurden. Zugesagte User Storys werden in Abschnitt 9.3.3.1. beschrieben.

9.4.3.4 Abhängigkeiten

Abhängigkeiten beschreiben die Beziehung und Interaktion zwischen verschiedenen Aufgaben innerhalb eines Projektes und können als zwingend oder nicht-zwingend, intern oder extern eingestuft werden, wie in Abschnitt 9.4.2.4. erörtert wird.

Es gibt verschiedene Möglichkeiten, um die Aufgaben und ihre Abhängigkeiten zu identifizieren, definieren und präsentieren. Zwei häufige verwendete Methoden beinhalten die Anwendung von Flussdiagrammen des Produkts und Gantt-Diagrammen.

9.5 Aufgaben Abschätzen

Dies ist ein optionaler Prozess, der die Erstellung von Aufgabenschätzungen beinhaltet, wenn das Scrum Team dies für sinnvoll hält. In diesem Prozess schätzt das Scrum Team den Aufwand ab, der für die Erledigung jeder Aufgabe in der Aufgabenliste erforderlich ist. Aufgabenschätzungen können entweder zu Beginn des Sprints für alle User Storys /Aufgaben, die für diesen Sprint relevant sind, oder für jede Aufgabe kurz bevor das Team mit der Arbeit an der jeweiligen User Story/Aufgabe beginnt, ermittelt werden. Die Schätzung kann mit denselben Methoden durchgeführt werden, die für den Prozess "User Storys schätzen" verwendet wurden.

Abbildung 9-14 zeigt Inputs, Hilfsmittel und Outputs für den *Aufgaben Abschätzen* Prozess.

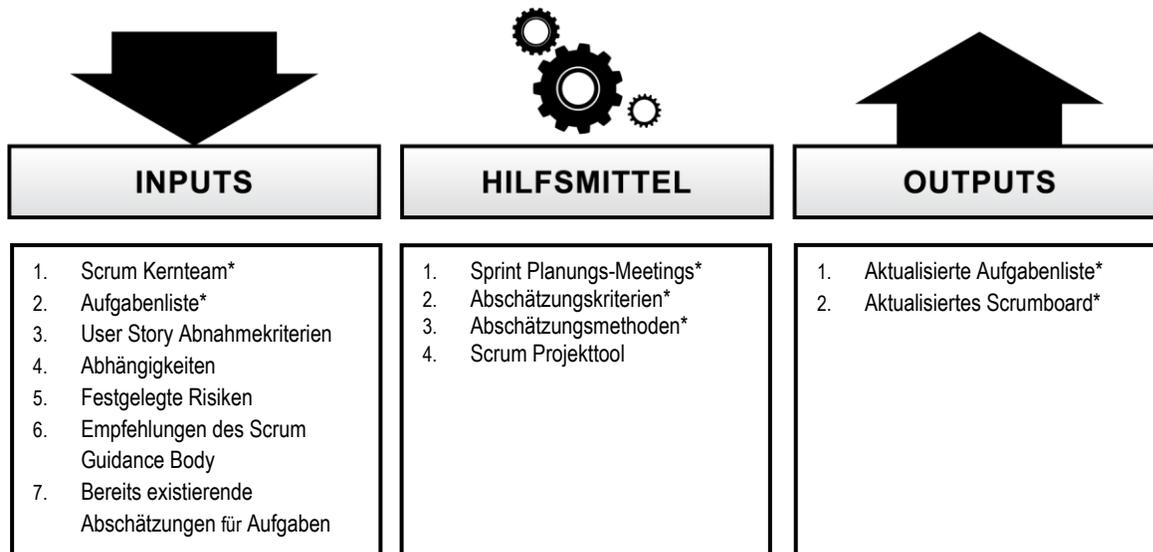


Abbildung 9-14: *Aufgaben Abschätzen* Prozess–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen (*) bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess

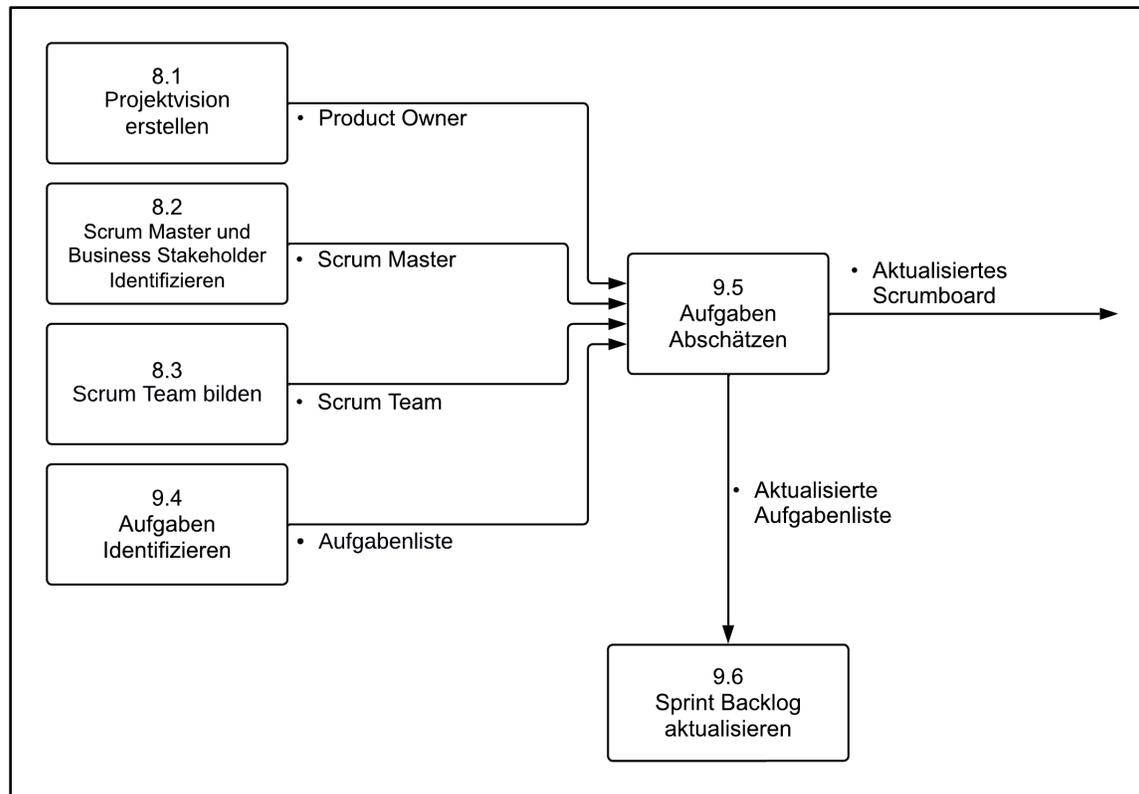


Abbildung 9-9: Aufgabenschätzung-Datenflussdiagramm

9.5.1 Inputs

9.5.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

9.5.1.2 Aufgabenliste*

Siehe Abschnitt 9.4.3.1.

9.5.1.3 User Story Abnahmekriterien

Siehe Abschnitt 9.1.3.2.

Das Scrum Team muss sicherstellen, dass die festgelegten Abnahmekriterien für die User Storys angemessen sind und Klarheit in Bezug auf die Anforderungen an das Scrum Team liefern.

Das Abnahme-Testen bezieht sich auf die Beurteilung der Fähigkeit der fertiggestellten zu liefernden Ergebnisse, die Abnahmekriterien zu erfüllen und stellt dem Product Owner, welcher über Genehmigung und Ablehnung entscheiden kann, Informationen zur Verfügung.

Bei der Erstellung der User Story Abnahmekriterien sollte Folgendes beachtet werden:

- Abnahmekriterien sollten nicht vage, mehrdeutig oder zu generell sein.
- Die definierten Abnahmekriterien sollten sicher stellen, dass das Team in der Lage ist, zu prüfen, ob sich die Ergebnisse an den Zielen und Zielsetzungen der Organisation des Sponsors orientieren.

9.5.1.4 Abhängigkeiten

Siehe Abschnitt 9.4.3.3

9.5.1.5 Identifizierte Risiken

Siehe Abschnitt 8.4.3.4.

9.5.1.6 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11.

Im Prozess *Aufgaben abschätzen* können die Empfehlungen des Scrum Guidance Body Informationen über Regeln, Regulierungen, Standards und die besten Praktiken, welche für die Aufgabenschätzung der Aufgabenliste benötigt werden, enthalten.

9.5.1.7 Bereits vorhandene Abschätzungen von Aufgaben

Es mag bereits vorhandene Aufgabenabschätzungen geben, die von den Mitgliedern des Scrum-Teams bei der Abschätzung von Aufgaben berücksichtigt werden können. Ähnliche Aufgaben wurden möglicherweise in demselben Projekt oder in anderen früheren Projekten bereits erledigt, und der Aufwand und die Zeit, die für die Erledigung dieser vergleichbaren Aufgaben benötigt wurden, können dem Scrum-Team helfen, während dieses Prozesses bessere Aufgabenabschätzungen zu erstellen. Experten, die in der Vergangenheit an ähnlichen Aufgaben gearbeitet haben, können ebenfalls einige Aufwandsschätzungen für Aufgaben abgeben. Es ist jedoch wichtig sicherzustellen, dass das Scrum-Team seine eigenen Abschätzungen für die Aufgaben erstellt, und sich nicht ausschließlich auf bereits vorhandene Schätzungen verlässt. Vorhandene Aufgabenschätzungen können dem Scrum Team auch dabei helfen, die Zusage an den Product Owner auf der Ebene der User Story neu zu bewerten.

9.5.2 Hilfsmittel

9.5.2.1 Sprint Planungs-Meetings *

Sprint Planungs-Meetings ermöglichen es dem Scrum Team, den menschlichen Aufwand und andere benötigte Ressourcen zu schätzen, um die Aufgaben in einem gegebenen Sprint umzusetzen. Die Mitglieder des Scrum Teams benutzen die Aufgabenliste, um die Dauer und den Aufwand für die User Storys, die im Sprint vervollständigt werden sollen, abzuschätzen. Einer der Hauptvorteile dieser Technik ist, dass es dem Team ermöglicht, eine gemeinsame Perspektive der User Storys und Bedürfnisse zu haben, sodass es zuverlässig den benötigten Aufwand schätzen kann.

Siehe auch Abschnitte 9.2.2.1, 9.3.2.1, 9.4.2.1 and 9.6.2.1.

9.5.2.2 Abschätzungskriterien*

Abschätzungskriterien können auf unterschiedliche Art und Weise formuliert werden, zwei verbreitete Beispiele sind Story Points und Idealzeit. So beschreibt beispielsweise Idealzeit normalerweise die Stundenzahl, welche ein Mitglied des Scrum Teams exklusiv an der Entwicklung der Lieferungen eines Projektes arbeitet. Dabei ist die Zeit, welche mit anderen Aktivitäten oder mit projektfernen Dingen verbracht wird nicht dazugerechnet. Abschätzungskriterien erleichtern es dem Scrum Team, den Aufwand zu schätzen und es ermöglicht ihm, Ineffizienz bei Bedarf zu bewerten und zu benennen.

9

9.5.2.3 Abschätzungsmethoden

Die gleichen Abschätzungsmethoden, die für die Abschätzung von User Storys verwendet werden, können auch auf Aufgaben angewendet werden. Weitere Informationen zu Schätzungsmethoden finden Sie in Abschnitt 9.2.2.1.

9.5.2.4 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

9.5.3 Outputs

9.5.3.1 Aktualisierte Aufgabenliste*

Die Aufgabenliste wird aktualisiert, um die geschätzten Aufwände einzubeziehen, die anhand der detaillierten Schätzungsaktivitäten im Prozess Aufgaben abschätzen ermittelt wurden. Es kann auch zu Neuabschätzungen kommen, die sich aus Änderungen im kollektiven Verständnis des Scrum-Teams von User Storys und Anforderungen ergeben. Der abgeschätzte Aufwand wird in Form der vom Team vereinbarten Schätzkriterien ausgedrückt. Typischerweise variiert die Genauigkeit der Abschätzungen mit den Fähigkeiten des Teams. Die aktualisierte Aufgabenliste wird vom Scrum Team während der Sprint Planungsmeetings zur Aktualisierung des Sprint Backlogs und zur Erstellung des Sprint Burndown Charts verwendet. Sie wird auch verwendet, um festzustellen, ob das Team seine Zusagen reduzieren muss oder ob es im Zuge der Sprint Planung für den anstehenden Sprint zusätzliche User Storys übernehmen kann.

9.5.3.2 Aktualisiertes Scrumboard*

Wenn Aufgaben abgeschätzt werden, wird das Scrumboard mit den Aufwandsabschätzungen aktualisiert. Weitere Informationen zum Scrumboard finden Sie in Abschnitten 9.3.3.3 und 9.4.3.2.

9.6 Sprint Backlog aktualisieren

9.6 Sprint Backlog aktualisieren — In diesem Prozess aktualisiert das Scrum Kernteam das Sprint Backlog mit den Aufgabendetails und, falls verfügbar, den Aufgabenschätzungen. Das aktualisierte Sprint Backlog wird in der Implementierungsphase verwendet, um den Fortschritt des Teams während des laufenden Sprints zu verfolgen.

Abbildung 9-16 zeigt Inputs, Hilfsmittel und Outputs für den *Sprint Backlog aktualisieren* Prozess.

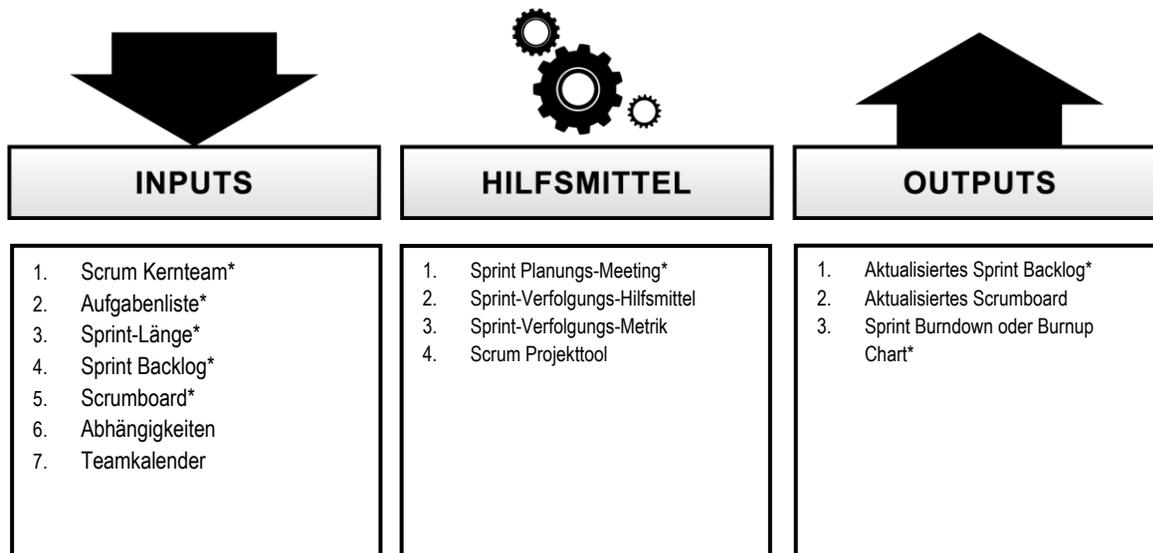


Abbildung 9-10: Sprint Backlog aktualisieren–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen (*) bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess

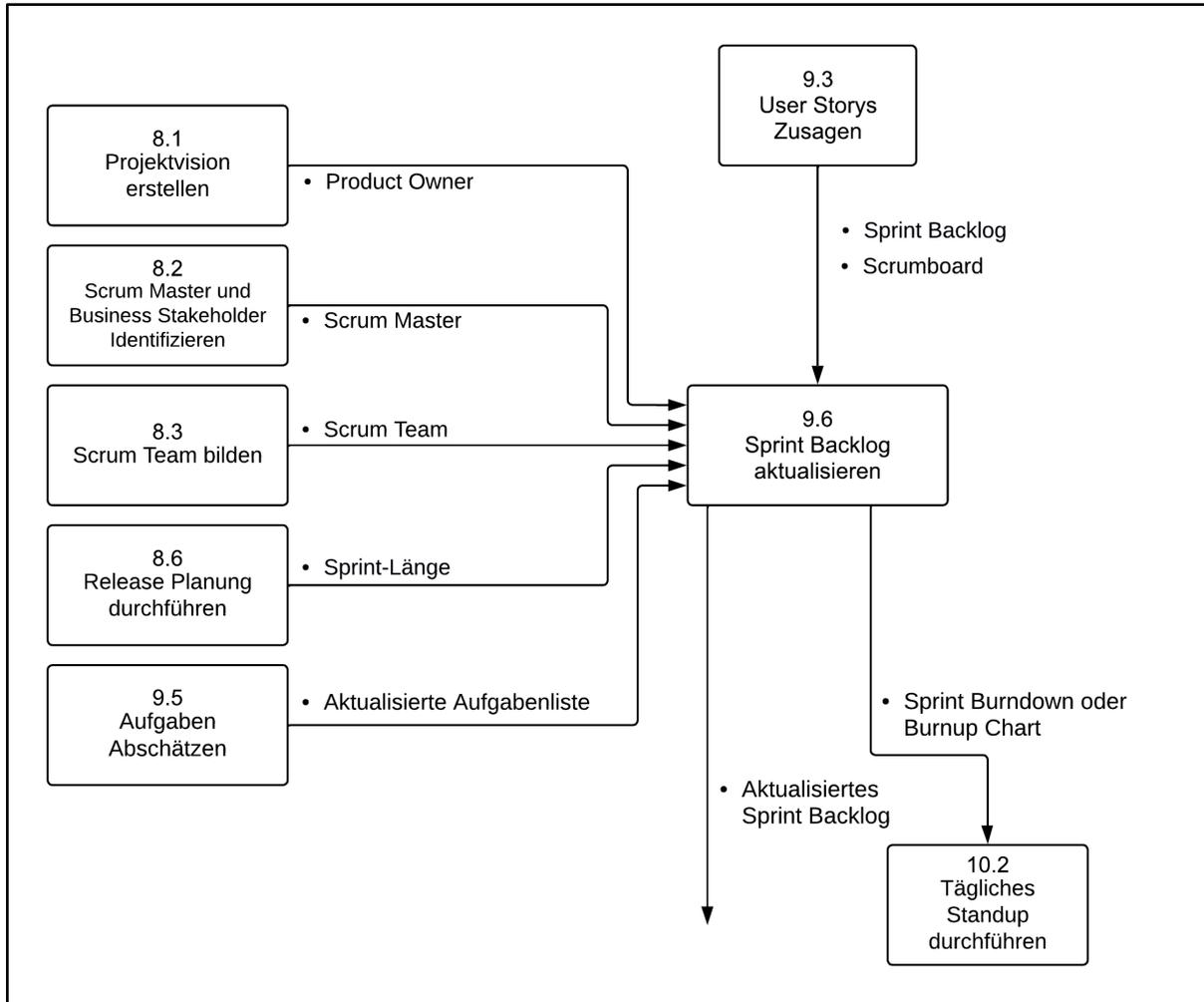


Abbildung 9-11: Sprint Backlog aktualisieren–Datenflussdiagramm

9.6.1 Inputs

9.6.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

9.6.1.2 Aufwands-abgeschätzte Aufgabenliste*

Siehe Abschnitt 9.5.3.1.

9.6.1.3 Sprint-Länge*

Siehe Abschnitt 8.6.3.2.

9.6.1.4 Abhängigkeiten

Siehe Abschnitt 9.4.3.3

9.6.1.5 Teamkalender

Ein Teamkalender enthält Informationen über die Verfügbarkeit der Teammitglieder einschließlich der Informationen, die sich auf Mitarbeiter-Urlaub, -Abwesenheiten, wichtige Ereignisse und Feiertage beziehen.

Eines der Hauptziele der Verwendung eines Teamkalenders ist es, aufzuzeichnen, woran jedes Teammitglied im Laufe des Projekts arbeitet. Das hilft dem Team nicht nur bei der effizienten Planung und Ausführung von Sprints sondern ebenfalls in der Angleichung der Sprints an die Veröffentlichungstermine.

9.6.2 Hilfsmittel

9.6.2.1 Sprint Planungs-Meeting*

Während der Sprint Planungs-Meetings werden User Storys fuer einen Sprint zugesagt und es werden Aufgaben vom Scrum Team identifiziert und abgeschätzt. Jedes Mitglied des Scrum Team benutzt die Aufwands-abgeschätzte Aufgabenliste, um je nach Fähigkeit und Erfahrung Aufgaben zu wählen, an denen sie im Sprint arbeiten wollen. Das Scrum Team erstellt ebenfalls das Sprint Backlog und den Sprint Burndown Chart, wozu es die User Storys und die Aufwands-abgeschätzte Aufgabenliste während des Sprint Planungs Meeting benutzt.

Siehe auch 9.2.2.1, 9.3.2.1, 9.4.2.1 und 9.5.2.1.

9.6.2.2 Sprint-Verfolgungs-Hilfsmittel

Es ist wichtig die Fortschritte des Sprint aufzuzeichnen und zu wissen, wo das Scrum Team in Bezug auf die Fertigstellung der Aufgaben im Sprint Backlog steht. Es gibt eine ganze Reihe von Hilfsmitteln, die dazu verwendet werden können, die Arbeit in einem Sprint zu verfolgen. Eines der Häufigsten ist ein Scrumboard, auch als Aufgaben-Board oder Fortschritts-Grafik bekannt. Die einfachste Version eines Scrumboards ist in drei Teile unterteilt: Zu erledigen (gelegentlich als noch nicht begonnene Arbeit bezeichnet), in Arbeit und fertiggestellte Arbeit. Post-its repräsentieren dabei jede Aufgabe oder User Story. Sie werden in eine passende Kategorie eingeordnet, um den Status der Arbeit zu reflektieren. Sie werden in die nächste Kategorie verschoben, wenn die Arbeit voranschreitet.

9.6.2.3 Sprint-Verfolgungs-Metrik

Einige Metriken, welche in Scrum Projekten genutzt werden können, beinhalten Geschwindigkeit, gelieferte Geschäftswerte und Storyanzahl.

Geschwindigkeit—repräsentiert die Anzahl der User Storys oder die Anzahl der Funktionen, die in einem einfachen Sprint geliefert werden.

Gelieferte Geschäftswerte—messen die Werte der User Storys, die aus Sicht des Geschäfts geliefert worden sind.

Storyanzahl—bezieht sich darauf, wie viele User Storys als Teil eines einzelnen Sprint geliefert werden. Es kann in Bezug auf einfaches Zählen oder gewichtetest Zählen genutzt werden.

9.6.3 Outputs

9.6.3.1 Aktualisiertes Sprint Backlog*

Das Scrum Kernteam aktualisiert das Sprint Backlog mit Details zu den Aufgaben, die mit den zugesagten User Storys im Sprint Backlog zusammenhängen. Falls verfügbar, werden auch die Aufgabenabschätzungen im Sprint Backlog aktualisiert. Das Sprint Backlog wird in der Implementierungsphase verwendet, um den Fortschritt des Teams während des Sprints zu verfolgen. Wenn das Sprint Backlog fertiggestellt und vom Scrum Team zugesagt wurde, sollten keine neuen User Storys hinzugefügt werden – Trotzdem kann es sein, dass Aufgaben, die vielleicht übersehen worden sind, hinzugefügt werden müssen. Wenn während eines Sprint eine neue Anforderung in Erscheinung tritt, werden Sie zum generellen Priorisierten Produkt Backlog hinzugefügt und in einen zukünftigen Sprint einbezogen.

9.6.3.2 Aktualisiertes Scrumboard*

Das Scrumboard wird aktualisiert, um die Informationen im aktualisierten Sprint Backlog wiederzugeben, einschließlich aller Aktualisierungen von Aufgaben, Aufgabenstatus und Aufgabenschätzungen, falls verfügbar.

9.6.3.3 Sprint Burndown oder Burnup Chart*

Burn-Charts (Burndown oder Burnup) werden verwendet, um den Fortschritt in einem Scrum-Projekt zu verfolgen. Ein Burndown Chart ist eine Grafik, die den Umfang der verbleibenden Arbeit im Verhältnis zur verbleibenden Zeit darstellt. Im Gegensatz zum Burn-down-Chart stellt ein Burnup-Chart dar, was im Verhältnis zur verbleibenden Zeit fertiggestellt wurde.

Burn Charts werden in der Implementierungsphase verwendet, um den Fortschritt des Scrum-Teams während eines Sprints zu verfolgen und einen frühen Hinweis darauf zu erhalten, ob das Team in der Lage sein wird, alle User Storys abzuschließen, die für diesen Sprint zugesagt wurden. Wenn die Teammitglieder glauben, dass sie nicht in der Lage sein werden, alle zugesagten User Storys fertigzustellen, können sie frühzeitig während des Sprints Maßnahmen ergreifen, um das bestmögliche Ergebnis zu erzielen.

Das anfängliche Sprint Burndown Chart zeigt, wie das Team die Arbeit zu erledigen gedenkt. Da das Team gerade eine Reihe von User Storys und die dazugehörigen Aufgaben für den aktuellen Sprint zugesagt hat und erwartet wird, dass das Team seine Zusagen einhält, zeigt das anfängliche geplante Burndown-Diagramm, dass am letzten Tag des Sprints keine Arbeit mehr zu erledigen sein wird. Das bedeutet, dass im Idealfall alle Arbeiten am letzten Tag erledigt sein werden. Das Burndown-Diagramm sollte vom Team am Ende eines jeden Tages aktualisiert werden, um den Fortschritt bei der Fertigstellung der Arbeit zu zeigen.

Ein Beispiel für ein Sprint-Burndown-Chart ist nachstehend in Abbildung 9-18 dargestellt:

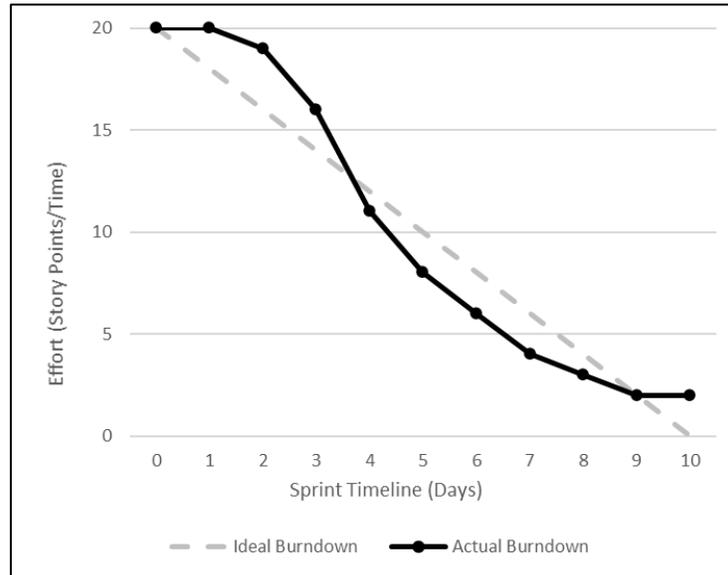


Abbildung 9-18: Sprint Burndown Chart

Ein Beispiel für ein Sprint-Burnup-Chart ist nachstehend in Abbildung 9-19 dargestellt:

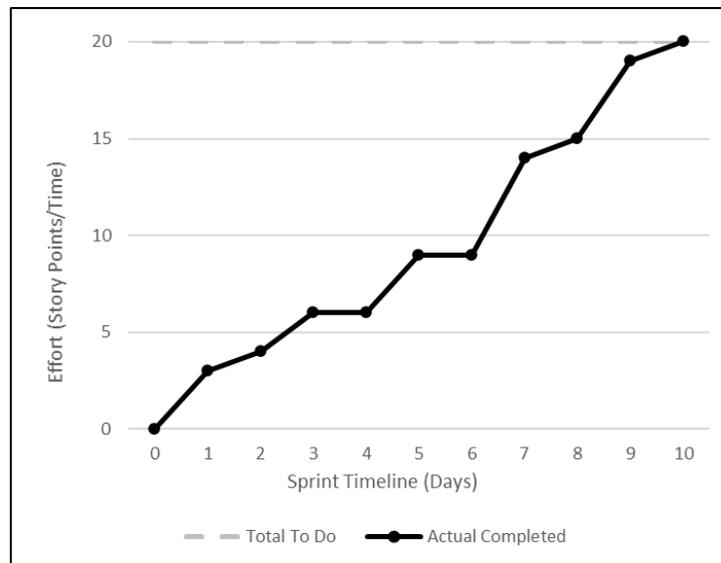


Abbildung 9-19: Sprint Burnup Chart

9.7 Datenflussdiagramm der Planungs- und Abschätzungs-Phase

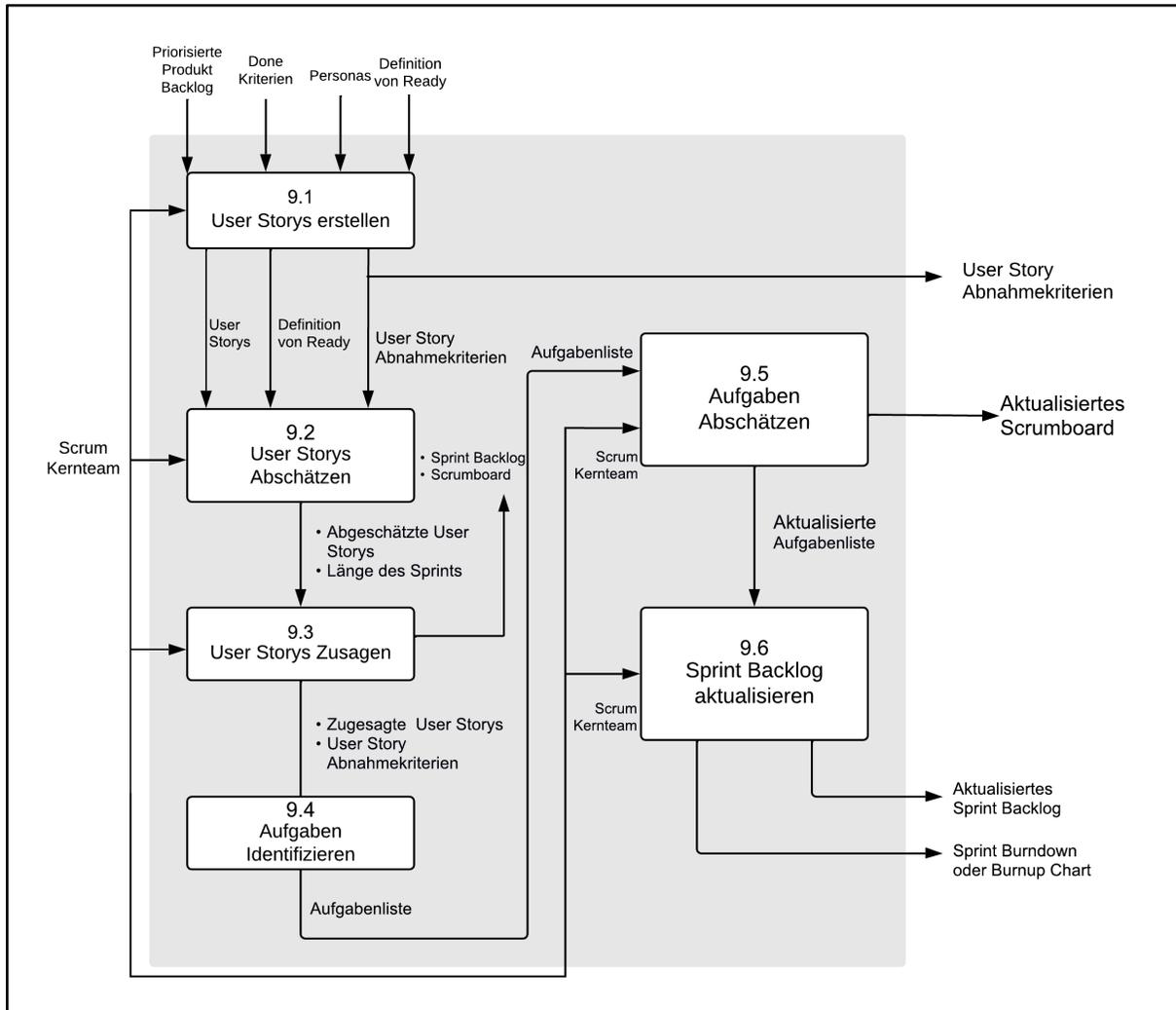


Abbildung 9-20 Planungs- und Abschätzungsphase-Datenflussdiagramm

10. IMPLEMENTIEREN

Die Implementierungsphase beinhaltet Prozesse, die mit der Ausführung der Aufgaben und Aktivitäten zur Erstellung des Produkts eines Projekts in Verbindung stehen. Diese Aktivitäten beinhalten das Erstellen der verschiedenen zu liefernden Ergebnisse, das Durchführen von täglichen Standup Meetings und die Pflege (z.B. Review, Feintuning und regelmäßige Updates) des Produkt Backlogs in regelmäßigen Abständen.

Implementieren, wie im *Scrum Body of Knowledge (SBOK®)* definiert, ist auf Folgendes anwendbar:

- Portfolios, Programme, und/oder Projekte in *jeder* Branche
- Produkte, Dienstleistungen, oder jegliche an Business Stakeholdern zu liefernde Ergebnisse
- Projekte jeglicher Größe oder Komplexität

Der Begriff "Produkt" in dieser Wissenssammlung kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann auf jedes Projekt in jeder Branche effektiv angewendet werden — von kleinen Projekten oder Teams mit lediglich sechs Team-Mitgliedern bis hin zu großen, komplexen Projekten mit mehreren Hundert Team-Mitgliedern.

Um die beste Anwendung des Scrum Rahmens zu ermöglichen, kennzeichnet dieses Kapitel Inputs, Hilfsmittel und Outputs für jeden Prozess entweder als „verbindlich“ oder „optional“. Inputs, Hilfsmittel und Outputs, die mit einem Sternchen (*) gekennzeichnet sind, sind verbindlich oder werden als kritisch fuer den Erfolg des Projekts angesehen, wohingegen jene ohne Sternchen optional sind.

Es wird empfohlen, dass das Scrum Team und alle Personen, die in den Scrum Rahmen und die Scrum Prozesse eingeführt werden, ihren Fokus vorwiegend auf die verbindlichen Inputs, Hilfsmittel und Outputs richten. Product Owner, Scrum Master und andere weiter fortgeschrittene Scrum Praktiker werden ein genaueres Verständnis des gesamten Kapitels erreichen wollen. Es ist außerdem wichtig, sich bewusst zu machen, dass obwohl alle Prozesse einzeln im *SBOK®* definiert sind, sie nicht unbedingt sequentiell oder getrennt angewendet werden. Manchmal empfiehlt es sich, je nach spezifischen Bedürfnissen eines jeden Projektes, einige Prozesse miteinander zu kombinieren.

Dieses Kapitel ist aus der Perspektive eines einzelnen Scrum Teams geschrieben worden, welches an einem Sprint arbeitet, um potenziell versandfähige Ergebnisse als Teil eines größeren Projektes, eines Programms oder eines Portfolios zu produzieren. Weitere Informationen, die sich auf den Gebrauch von Scrum für große Projekte beziehen, sind in Kapitel 13 zu finden, die Skalierung von Scrum für Programme und Portfolios befinden sich in Kapitel 14.

Die Implementierung ist die zweite der drei Phasen, die sich in jedem Sprint wiederholen. Diese Phase beginnt, nachdem die Sprintplanung abgeschlossen ist. Sie ist der Kern eines jeden Scrum-Projekts, in dem der Großteil der Arbeit geleistet wird.

Das Scrum Team erstellt mit der Unterstützung des Scrum Masters die zu Liefernden Ergebnisse, die mit den zugesagten User Storys im Zusammenhang stehen, indem es die Aufgaben, die das Team in der vorherigen Phase identifiziert hat, bearbeitet und abschließt.

Während das Scrum Team die zu Liefernden Ergebnisse für den Sprint erstellt, aktualisiert und verfeinert der Product Owner das priorisierte Product Backlog, um es bei Änderungen der Anforderungen und/oder Prioritäten auf dem neuesten Stand zu halten und um sicherzustellen, dass die User Storys, die der Product Owner im nächsten Sprint für das Team vorsehen möchte, bereit für die Zusage sind.

Es ist auch wichtig zu wissen, dass, obwohl alle Phasen und Prozesse im SBOK® Guide eindeutig definiert sind, sie nicht unbedingt sequenziell oder separat durchgeführt werden. Je nach den spezifischen Anforderungen des jeweiligen Projekts kann es manchmal sinnvoller sein, dass sich einige Phasen und/oder Prozesse überlappen.

Abbildung 10-1 gibt einen Überblick über die Prozesse der Implementierungsphase, welche nachstehend gelistet sind:

10.1 Zu liefernde Ergebnisse erzeugen— In diesem Prozess erstellt das Scrum Team die zu Liefernden Ergebnisse des Sprints, indem es an den Aufgaben im Sprint Backlog arbeitet. Dies ist der Prozess, in dem das Scrum Team und der Scrum Master die meiste Zeit investieren. Das Team wird vom Scrum Master unterstützt, der Meetings für das Team moderiert, Hindernisse adressiert, mit denen das Team konfrontiert wird, und alles tut, was er/sie tun kann, damit sich die Mitglieder des Scrum Teams auf die Erstellung der zu Liefernden Ergebnisse des Sprints konzentrieren können.

Das Scrum Team verwendet ein Scrumboard, um seinen Fortschritt während des Sprints zu verfolgen. Das Scrum Team nutzt die Informationen über seinen Fortschritt, um einen guten Hinweis auf seine Fähigkeit zu erhalten, gemäß seiner Zusage zu liefern und, falls erforderlich, Maßnahmen zu ergreifen, um das hochwertigste Ergebnis des Sprints zu sichern, das unter den gegebenen Umständen möglich ist.

10.2 Tägliches Standup durchführen—In diesem Prozess wird jeden Tag ein hoch konzentriertes Meeting durchgeführt wird. Man spricht von diesem Meeting als Tägliches Standup Meeting. Dieses time-boxed Meeting ist ein Forum für das Scrum Team, bei dem sich die Team Mitglieder gegenseitig über Ihren Fortschritt und jegliche Hindernisse informieren, denen sie ggfls. gegenüberstehen.

10.3 Priorisiertes Produkt Backlog verfeinern—Das ist ein Prozess, in dem der Product Owner das Priorisierte Produkt Backlog ständig aktualisiert und pflegt. Ein Priorisiertes Produkt Backlog Review Meeting kann abgehalten werden. In ihm werden alle Veränderungen oder Updates des Backlog besprochen und entsprechend in das Priorisierte Produkt Backlog eingetragen.

Um das priorisierte Produkt Backlog bei Änderungen der Anforderungen und/oder Prioritäten auf dem neuesten Stand zu halten, arbeitet der Product Owner kontinuierlich mit dem Kunden und anderen Business Stakeholdern des Unternehmens zusammen, um alle Änderungen ihrer Bedürfnisse zu erfassen und zu verstehen.

Um sicherzustellen, dass die User Storys, für die der Product Owner vom Team im nächsten Sprint eine Zusage erhalten möchte, bereit sind, verfeinert der Product Owner die bestehenden Epics und User Storys im priorisierten Produkt Backlog und stellt sicher, dass die User Storys die Definition von Bereit erfüllen.

Als Teil der Verfeinerung des priorisierten Produkt Backlogs arbeitet der Product Owner auch mit dem Scrum Team zusammen, um Feedback und Fragen in Bezug auf die Aktualisierungen im priorisierten Produkt Backlog zu erhalten, möglicherweise einschließlich Abschätzungen.

Wenn Änderungen der Anforderungen und/oder des Gesamtfortschritts des Scrum-Teams Änderungen des Release-Plans und/oder der geschäftlichen Rechtfertigung erfordern, nimmt der Product Owner diese Änderungen ebenfalls während dieses Prozesses vor.

Dies ist der Prozess, in dem der Product Owner die meiste Zeit investieren wird.

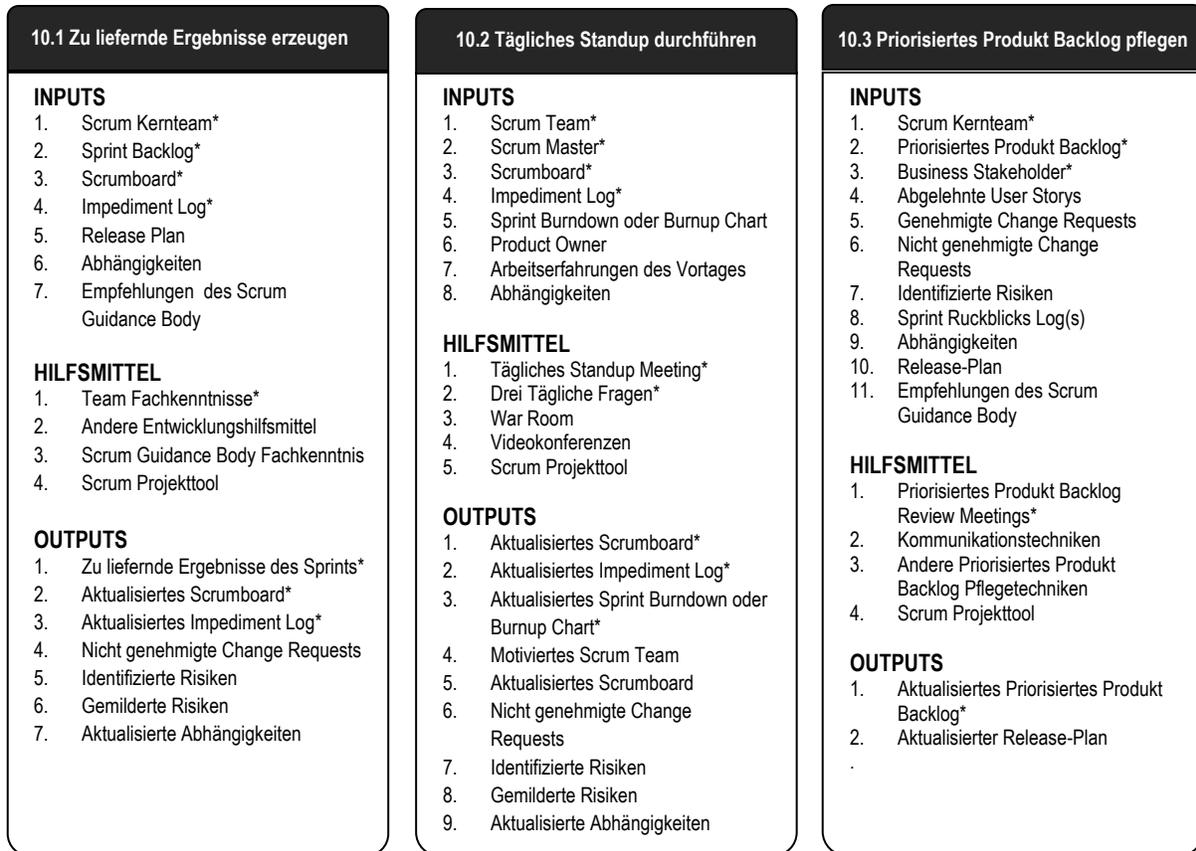


Abbildung 10-1: Implementierungsübersicht

Abbildung 10-2 unten zeigt die verbindlichen Inputs, Hilfsmittel und Outputs für Prozesse in der Implementierungsphase.

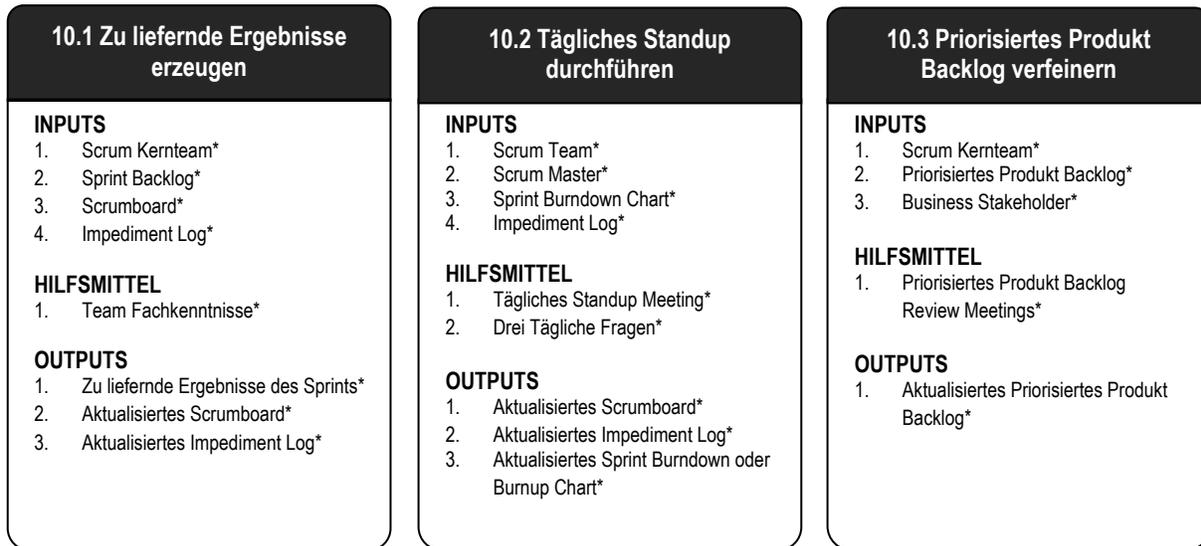


Abbildung 10-2: Implementierungsübersicht (Kernpunkte)

10.1 Zu liefernde Ergebnisse erzeugen

In diesem Prozess erstellt das Scrum Team die zu Liefernden Ergebnisse des Sprints, indem es an den Aufgaben im Sprint Backlog arbeitet. Dies ist der Prozess, in dem das Scrum Team und der Scrum Master die meiste Zeit investieren. Das Team wird vom Scrum Master unterstützt, der Meetings für das Team moderiert, Hindernisse adressiert, mit denen das Team konfrontiert wird, und alles tut, was er/sie tun kann, damit sich die Mitglieder des Scrum Teams auf die Erstellung der zu Liefernden Ergebnisse des Sprints konzentrieren können.

Das Scrum Team verwendet ein Scrumboard, um seinen Fortschritt während des Sprints zu verfolgen. Das Scrum Team nutzt die Informationen über seinen Fortschritt, um einen guten Hinweis auf seine Fähigkeit zu erhalten, gemäß seiner Zusage zu liefern und, falls erforderlich, Maßnahmen zu ergreifen, um das hochwertigste Ergebnis des Sprints zu sichern, das unter den gegebenen Umständen möglich ist.

Abbildung 10-3 zeigt Inputs, Hilfsmittel und Outputs für den *Zu liefernde Ergebnisse erzeugen* Prozess.

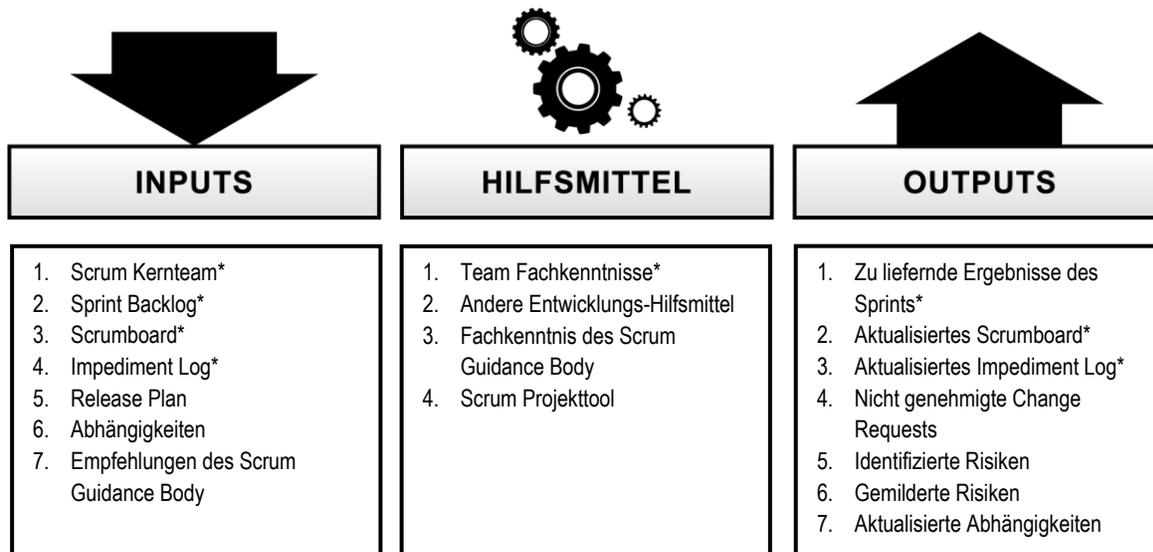


Abbildung 10-3: Zu liefernde Ergebnisse erzeugen–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen (*) bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess

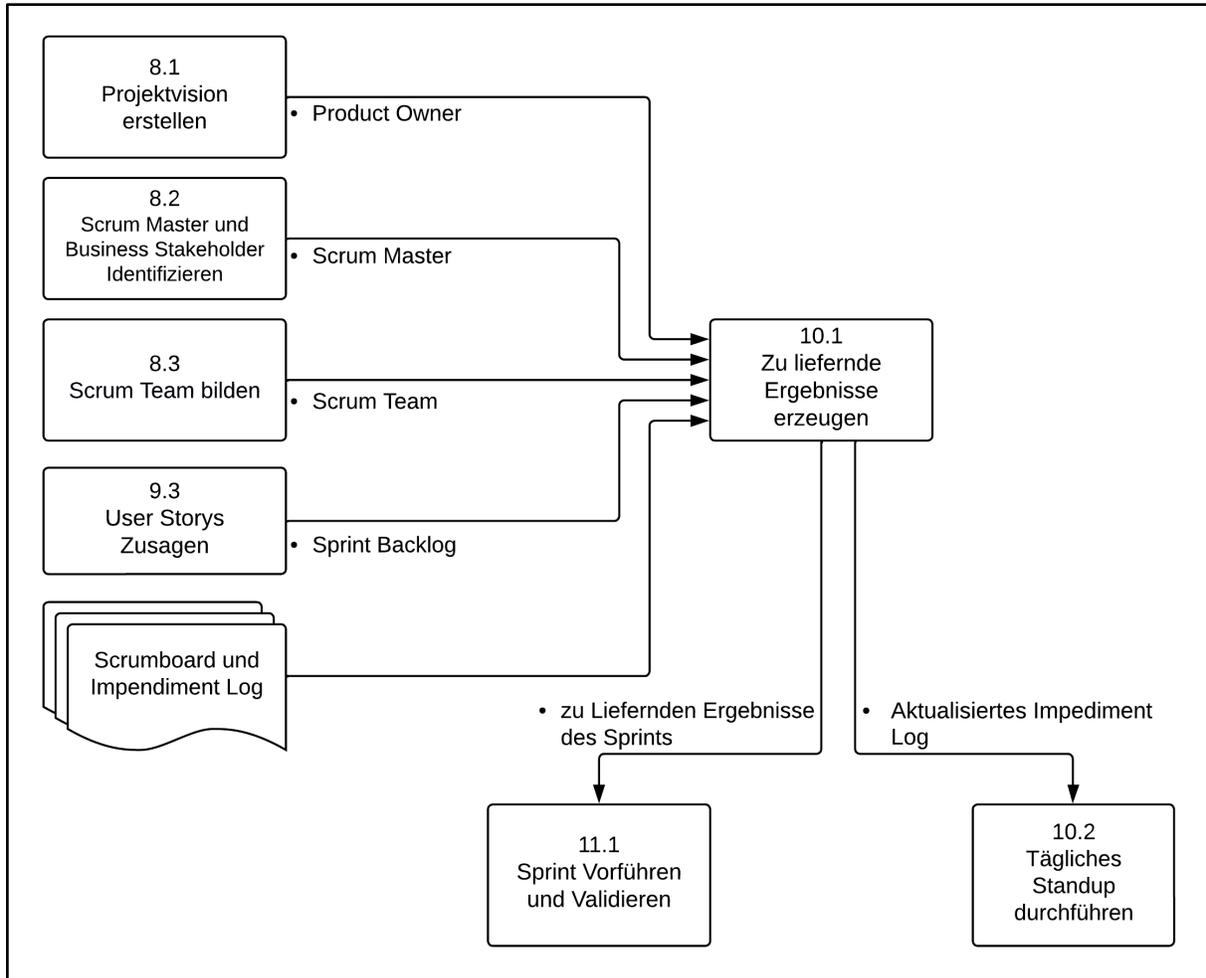


Abbildung 10-4: Zu liefernde Ergebnisse erzeugen–Datenflussdiagramm

10.1.1 Inputs

10.1.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

10.1.1.2 Sprint Backlog*

Siehe Abschnitt 9.5.3.1.

10.1.1.3 Scrumboard*

Siehe Abschnitte 9.3.3.3 und 9.4.3.2.

Die Transparenz von Scrum kommt von offen sichtbaren Hilfsmitteln zur Informationsverbreitung wie dem Scrumboard, welches den Teamfortschritt anzeigt. Das Scrum-Team sollte das Scrumboard nach Bedarf aktualisieren, damit das Scrumboard genaue visuelle Informationen und Kontrolle über die Arbeit bietet, die vereinbart und vom Team zugesagt läuft. Das Team verwendet das Scrumboard, das in der Planungs- und Abschätzungsphase aus dem Sprint Backlog erstellt wurde, das zu Beginn des Sprints alle Aufgaben in der Spalte "Zu erledigen" enthält. Die Mitglieder des Scrum-Teams überprüfen die User Storys und Aufgaben im Scrumboard täglich und verschieben die Aufgaben in die Spalten "In Arbeit" und "Erledigt", wenn die Arbeit voranschreitet. Je nach Arbeitsablauf des Scrum-Teams bei der Erstellung von zu liefernden Ergebnissen kann eine zusätzliche Spalte (z. B. "Testen") zum Scrumboard hinzugefügt werden.

Sobald das Team Aufgaben hinzufügt/aktualisiert und Aufgaben zur Bearbeitung zuweist, wird das Scrumboard ständig mit den zusätzlichen Aufgaben und dem Status der Aufgaben aktualisiert. Abbildung 10-5 zeigt beispielsweise, dass alle Aufgaben für die User Story 1 abgeschlossen sind; das Team arbeitet jedoch derzeit an einigen Aufgaben für die User Storys 2 und 3. User Story 4 wurde in Aufgaben zerlegt, aber das Scrum-Team hat noch nicht mit der Arbeit an den Aufgaben für diese User Story begonnen.

Storys	Unerledigt	In Arbeit	Im Test	Erledigt
1			■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■
2		■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■
3	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■	
4	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■	

Abbildung 10-5: Scrumboard mit Aufgaben in „Zu erledigen“, „In Arbeit“ und „Erledigt“

Um sicherzustellen, dass die Mitglieder des Scrum-Teams die Verantwortung für ihre Arbeit übernehmen, wird empfohlen, dass das Scrum-Teammitglied, das an einer bestimmten Aufgabe arbeitet, die Aufgabe von "Zu erledigen" auf "In Arbeit" verschiebt und der Aufgabe seinen/ihren Namen hinzufügt. Dadurch weist sich die Person, die für die Erledigung der Aufgabe die Verantwortung übernimmt, die Aufgabe selbst zu. Außerdem sollte nur ein Mitglied des Scrum-Teams für die Erledigung jeder Aufgabe verantwortlich sein. User Stories sollten also so in Aufgaben unterteilt werden, dass nur eine Person für eine Aufgabe bis zu deren Abschluss verantwortlich sein kann.

Wenn das Scrum-Team mit der Arbeit an einer User Story beginnt, hat es möglicherweise ein besseres Verständnis für die Aufgaben gewonnen, die zur Fertigstellung der User Story erforderlich sind. Dies kann dazu führen, dass Aufgaben hinzugefügt, aktualisiert oder vom Scrumboard entfernt werden müssen, je nach Entscheidung des Scrum-Teams.

Wenn alle Aufgaben für eine User Story abgeschlossen sind (z.B. wie in User Story 1 in der obigen Abbildung), wird die User Story vom Scrum Team als erledigt betrachtet. Erledigte User Storys stehen dann dem Product Owner zur Überprüfung zur Verfügung und können entweder genehmigt oder abgelehnt werden. Die Überprüfung der User Storys durch den Product Owner kann entweder nach der Fertigstellung der User Storys oder während des Prozesses Sprint vorführen und validieren durch den Product Owner erfolgen. Wenn eine User Story vom Product Owner genehmigt wird, dann wird diese User Story vom Scrum Team als "Done" betrachtet (und es muss keine weitere Arbeit vom Scrum Team an dieser User Story geleistet werden).

Wenn der Product Owner eine User Story ablehnt, muss der Product Owner angeben, warum die User Story abgelehnt wurde (d.h. welche Elemente der Abnahmekriterien und/oder Done-Kriterien nicht erfüllt wurden).

Abhängig von der verbleibenden Zeit im Sprint gibt es nach der Ablehnung einer User Story und der Angabe der Gründe des Product Owners für die Ablehnung der User Story zwei Möglichkeiten für das Scrum Team:

- Die abgelehnte User Story im aktuellen Sprint überarbeiten (basierend auf dem vom Product Owner bereitgestellten Input). Wenn alle für die User Story erforderlichen Aufgaben abgeschlossen sind, kann das Scrum Team die User Story dem Product Owner im selben Sprint dann erneut zur Genehmigung vorlegen.
- Die abgelehnte User Story im aktuellen Sprint nicht überarbeiten. In diesem Fall geht die User Story zurück in das priorisierte Product Backlog, damit sie einem anderen Sprint zugeordnet werden kann. Die User Story kann erneut demselben Scrum Team zugewiesen werden, oder ein anderes Scrum Team kann in einem zukünftigen Sprint die Verantwortung für diese User Story übernehmen.

Das Scrumboard kann manuell auf Papier oder auf einem großen Whiteboard geführt werden, es kann aber auch elektronisch in einer Tabellenkalkulation oder mit einem Scrum Projekttool geführt werden. Ein Scrumboard ist für die Dauer eines Sprints gültig. Das Scrum Team wird im nächsten Sprint ein neues Scrumboard erstellen.

10.1.1.4 Impediment Log*

Unter Impedimenten versteht man sämtliche Hindernisse oder Hemmnisse, welche die Produktivität des Scrum Teams verringern. Impedimente müssen identifiziert, gelöst und beseitigt werden, wenn das Team weiterhin effektiv arbeiten soll. Impediments können intern, also innerhalb des Teams sein, wie beispielsweise ein ineffizienter Arbeitsfluss oder mangelnde Kommunikation. Sie können aber auch extern sein. Als Beispiele externer Impedimente können Fragen hinsichtlich Softwarelizenzen, oder unwichtige Dokumentationsanforderungen genannt werden. Der Scrum Rahmen, mit seiner inhärenten Transparenz, ermöglicht die einfache Identifizierung der Impedimente. Das Versäumnis, Impedimente zu identifizieren oder sie anzugehen kann sehr kostspielig sein. Impedimente sollten formal durch den Scrum Master in einem Impediment Log erfasst werden und können während der täglichen Standup Meetings und den Sprint Review Meetings wie angemessen besprochen werden.

10.1.1.5 Release Plan

Siehe Abschnitt 8.6.3.1.

10.1.1.6 Abhängigkeiten

Siehe Abschnitt 9.3.3.3.

10.1.1.7 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11.

Während des Prozesses *Zu liefernde Ergebnisse erzeugen*, können die Empfehlungen des Scrum Guidance Body die besten Praktiken enthalten, um effektiv zu liefernde Ergebnisse zu erzeugen. Darin inbegriffen sind bevorzugte Methoden, um Reviews, Tests, Dokumentationen etc. durchzuführen.

10.1.2 Hilfsmittel

10.1.2.1 Team Fachkenntnisse*

Team Fachkenntnisse beziehen sich auf das Know-how der Scrum Team-Mitglieder, die User Storys und Aufgaben im Sprint Backlog gut genug zu verstehen, um die letztendlich zu liefernden Ergebnisse zu erzeugen. Teamkompetenzen werden verwendet, um die Inputs zu bewerten, welche benötigt werden, um die geplanten Arbeiten des Projekts auszuführen. Die Bewertung und Expertise werden auf alle technischen und managementbezogenen Aspekte des Projekts während des *Zu liefernde Ergebnisse erzeugen* Prozesses bezogen. Die Scrum Teammitglieder haben die Autorität und Verantwortung, die beste Vorgehensweise zu bestimmen, um die Einträge des Priorisierten Produkt Backlogs in fertige Produkte zu verwandeln. Dabei sind keine Business Stakeholdern außerhalb des Teams involviert. Zusätzliche Expertise ist im Scrum Guidance Body verfügbar, falls sie benötigt wird.

10.1.2.2 Andere Entwicklungshilfsmittel

Basierend auf den spezifischen Anforderungen eines Projekts und auf Besonderheiten der Industrie können auch andere Entwicklungs Hilfsmittel benutzt werden.

1. Refactoring

Refactoring ist ein Begriff aus der Softwareentwicklung. Das Ziel dieser Technik ist, die Wartbarkeit des vorhandenen Codes zu verbessern und ihn einfacher, präziser und flexibler zu machen. Refactoring bedeutet das Design des vorliegenden Codes zu verbessern, ohne das Verhalten des Codes zu verändern. Es beinhaltet Folgendes:

- Die Beseitigung von sich wiederholendem und überflüssigem Code
- Zerlegung von Methoden und Funktionen in kleinere Routinen
- Klare Definition von Variablen und Methodennamen
- Vereinfachung des Codedesigns
- Den Code besser verständlich und leichter anpassbar machen

Regelmäßiges Refactoring optimiert das Code Design in kleinen Schritten, die über einen längeren Zeitraum ausgeführt werden. Letztendlich resultiert Refactoring in saubererem und besser pflegbarem Codes, während gleichzeitig alle Funktionalitäten erhalten werden.

2. Entwurfsmuster

Entwurfsmuster bieten eine formale Möglichkeit, Lösungen von Designproblemen in einem bestimmten Fachgebiet festzuhalten. Diese Muster halten sowohl den Prozess als auch die tatsächliche Lösung fest und können im späteren Verlauf erneut genutzt werden, um Entscheidungsfindung und Produktivität zu verbessern.

10.1.2.3 Fachkenntnis des Scrum Guidance Body

Siehe Abschnitt 8.4.2.7.

Im Prozess *Zu liefernde Ergebnisse erzeugen* kann sich die Fachkenntnis des Scrum Guidance Body auf dokumentierte Regeln und Verordnungen, Entwicklungsrichtlinien oder Standards und beste Praktiken (z.B. Hinweise zur Durchführung von Reviews und Tests) beziehen. Es kann auch ein Team aus Fachexperten zusammengestellt werden, welche das Scrum Team beim Erstellen von zu liefernden Ergebnissen unterstützen kann. Dieses Team könnte aus führenden Architekten, erfahrenen Entwicklern, Sicherheitsexperten oder anderen erfahrenen Personen bestehen.

10

10.1.2.4 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

10.1.3 Outputs

10.1.3.1 Zu liefernde Ergebnisse des Sprints*

Am Ende eines jeden Sprint ist ein Produktinkrement oder zu lieferndes Ergebnis fertiggestellt. Die Lieferung sollte alle definierten Eigenschaften und Funktionalitäten, welche in den User Storys des Sprint definiert sind, enthalten. Sie sollten auch erfolgreich getestet worden sein.

10.1.3.2 Aktualisiertes Scrumboard*

Das Scrumboard wird regelmäßig aktualisiert während das Team Aufgaben erfüllt. Am Ende des Sprint wird das Scrumboard wieder zurückgesetzt oder abgewischt und für den nächsten Sprint wird ein neues Scrumboard erstellt.

10.1.3.3 Aktualisiertes Impediment Log*

Siehe Abschnitt 10.1.1.4.

10.1.3.4 Nicht genehmigte Change Requests

Siehe Abschnitt 8.4.1.6.

10.1.3.5 Identifizierte Risiken

Siehe Abschnitt 8.4.3.4.

10.1.3.6 Gemilderte Risiken

Das Scrum Team führt Arbeiten aus, um die den User Storys im Produkt Backlog entsprechenden zu liefernden Ergebnisse zu erstellen. Währenddessen führt es mildernde Aktionen aus, die definiert worden sind, um zuvor identifizierte Risiken anzusprechen. Während des Prozesses *Zu liefernde Ergebnisse erzeugen* dokumentiert das Team alle neu identifizierten Risiken und mildernden Schritte, welche durchgeführt worden sind. Die Aufzeichnung der Projekt Risiken ist ein lebendiges Dokument, welches im Laufe des Projektes dauerhaft vom Team aktualisiert wird, um den aktuellen Status aller Risiken reflektieren zu können.

Zusätzliche Informationen über den Umgang mit Risiken sind in Abschnitt 7.4.3 beschrieben.

10.1.3.7 Aktualisiertes Abhängigkeiten

Siehe Abschnitt 9.3.3.3.

10.2 Tägliches Standup durchführen

In diesem Prozess wird jeden Tag ein hoch konzentriertes Meeting durchgeführt. Man spricht von diesem Meeting als Tägliches Standup Meeting. Dieses time-boxed Meeting ist ein Forum für das Scrum Team, bei dem sich die Team Mitglieder gegenseitig über Ihren Fortschritt und jegliche Hindernisse informieren, denen sie ggfls. gegenüberstehen.

Abbildung 10-6 zeigt Inputs, Hilfsmittel und Outputs für den *Tägliches Standup durchführen* Prozess.

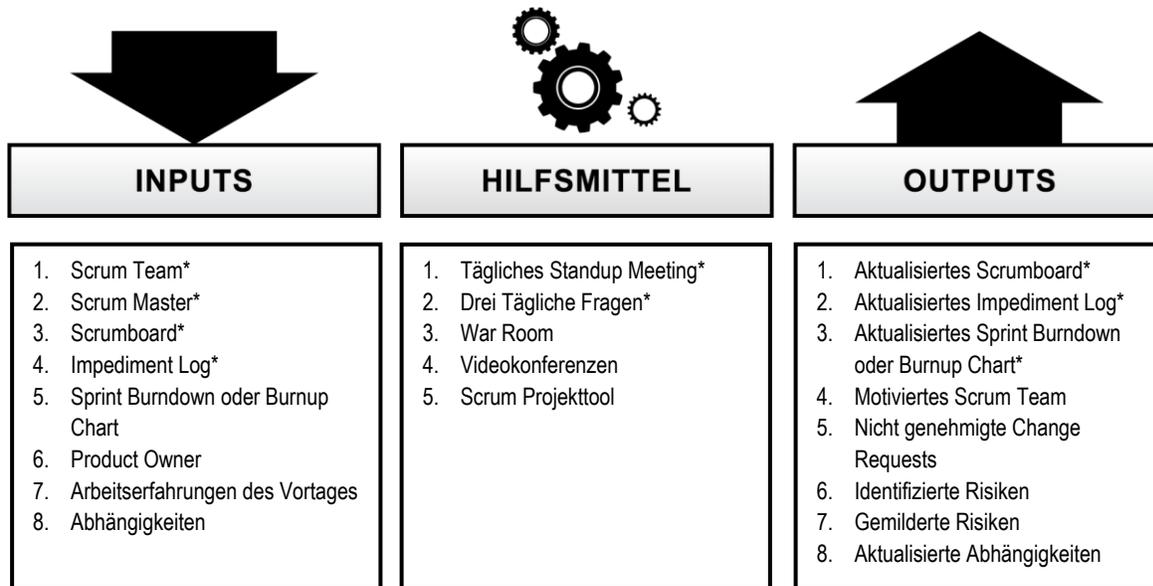


Abbildung 10-6: Tägliches Standup durchführen–Inputs, Hilfsmittel, and Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

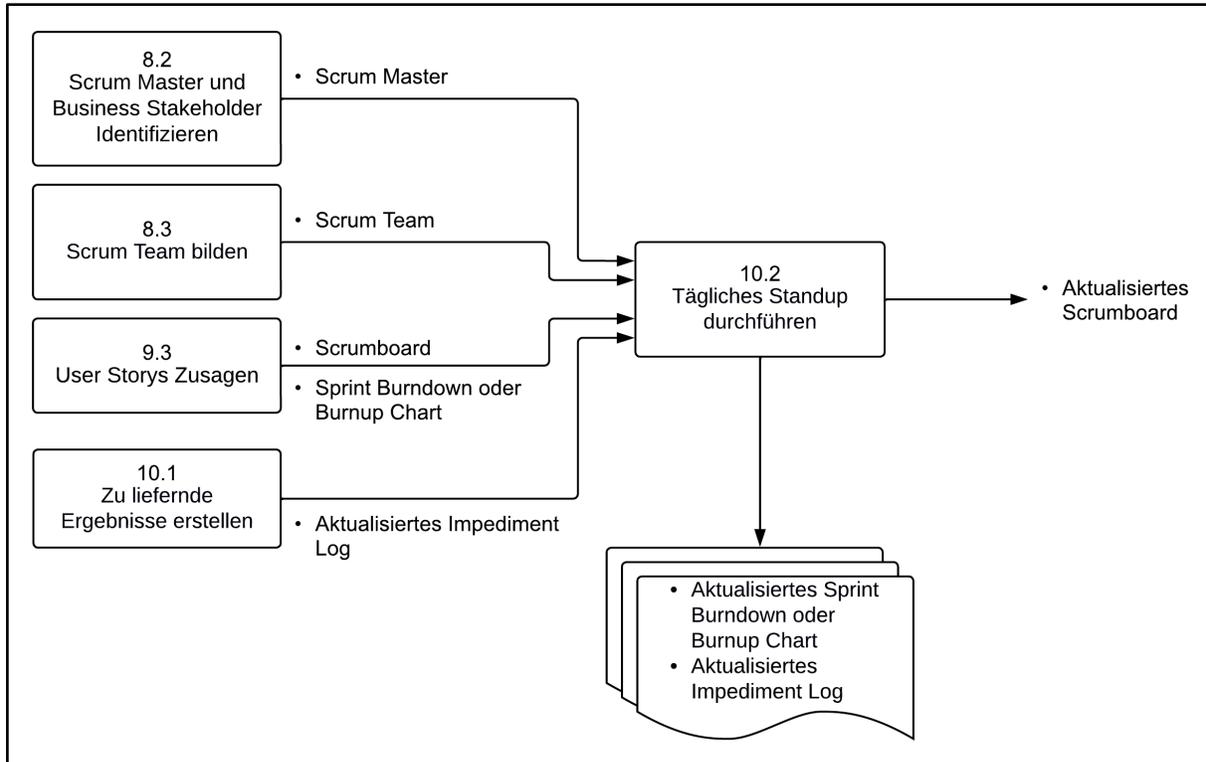


Abbildung 10-7: Tägliches Standup durchführen–Datenflussdiagramm

10.2.1 Inputs

10.2.1.1 Scrum Team*

Siehe Abschnitt 8.3.3.1.

10.2.1.2 Scrum Master*

Siehe Abschnitt 8.2.3.1.

10.2.1.3 Scrumboard*

Siehe Abschnitte 9.3.3.3, 9.4.3.2 und 10.1.1.3

10.2.1.4 Impediment Log*

Siehe Abschnitt 10.1.1.4.

10.2.1.5 Sprint Burndown oder Burnup Chart*

Siehe Abschnitt 9.5.3.2.

10.2.1.6 Product Owner

Siehe Abschnitt 8.1.3.1.

10.2.1.7 Arbeitserfahrungen des Vortages

Die Scrum Teammitglieder teilen den anderen Teammitgliedern beim Täglichen Standup Meeting Statusänderungen mit. Dieses Treffen wird als Standup bezeichnet, weil die Mitglieder während des Meetings stehen. Die Teammitglieder besprechen Erreichtes und Erfahrungen des Vortages. Diese Erfahrung ist ein wichtiger Input für die täglichen Standup Meetings.

10

10.2.1.8 Abhängigkeiten

Siehe Abschnitt 9.3.3.3.

10.2.2 Hilfsmittel

10.2.2.1 Tägliches Standup Meeting*

Unter dem Täglichen Standup Meeting versteht man ein kurzes tägliches Meeting mit einer Time-Box von 15 Minuten. Die Teammitglieder treffen zusammen, um von ihren Fortschritten im Sprint zu berichten und die Aktivitäten des Tages zu planen. Die Dauer des Treffens ist sehr kurz und alle Mitglieder des Scrum Teams müssen anwesend sein. Das Meeting wird jedoch nicht abgesagt oder verschoben, wenn ein oder mehrere Teammitglieder nicht in der Lage sind, teilzunehmen. Das Daily Standup Meeting wird vom Scrum Team selbst geleitet, während der Scrum Master das Meeting je nach Bedarf unterstützt.

Bei diesem Meeting gibt jedes Mitglied des Scrum Teams Antworten zu den Drei täglichen Fragen, wie in Abschnitt 10.2.2.2 bereits beschrieben wurde. Diskussionen zwischen dem Team und dem Scrum Master werden ermutigt. Solche Diskussionen finden aber nach dem Meeting statt, um sicherzustellen, dass das Tägliche Standup Meeting kurz bleibt.

10.2.2.2 Drei tägliche Fragen*

Im täglichen Standup Meeting ermöglicht der Scrum Master jedem Mitglied des Scrum Teams, von seinem Fortschritt zu berichten, indem es die folgenden drei Fragen beantwortet:

- Was habe ich seit dem letzten Meeting gemacht?
- Was plane ich bis zum nächsten Meeting zu machen?
- Vor welchen Impedimenten oder Hemmnissen (falls vorhanden) stehe ich zurzeit?

Durch das Fokussieren auf diese drei Fragen kann das gesamte Team ein klares Verständnis über den Arbeitsstatus erlangen. Gelegentlich werden dabei auch andere Dinge besprochen. Dies sollte aber die Ausnahme bleiben, da das Meeting Time-Boxed ist.

Es wird wärmstens empfohlen, dass die ersten beiden Fragen durch die Teammitglieder, wenn möglich, quantitativ beantwortet werden und auf lange qualitative Antworten verzichtet wird.

10.2.2.3 War Room

In Scrum ist es besser für das Team, wenn es zusammen arbeitet und sich alle Teammitglieder am gleichen Ort befinden. Die Bezeichnung, welche normalerweise benutzt wird, um diesen Platz zu beschreiben, ist „War Room“. Normalerweise ist er so konzipiert, dass Teammitglieder dort arbeiten, sich frei bewegen und außerdem leicht miteinander kommunizieren können, weil sie sich in unmittelbarer Nähe zueinander befinden. Um den Arbeitsfluss, Zusammenarbeit und Lösung von Problemen zu unterstützen, werden normalerweise Indexkarten, Post-its und andere einfache und haptische Hilfsmittel verwendet.

Aufgrund von Teamunterhaltungen ist es im Raum manchmal laut, aber diese Unterhaltungen tragen zum Fortschritt des Teams bei. Ein guter War Room hat keine Kabinen und ermöglicht dem gesamten Team, von Angesicht zu Angesicht zu kommunizieren. Dies hilft dem Aufbau des Teams und der Offenheit. Der War Room eignet sich hervorragend, um das tägliche Standup Meeting abzuhalten.

Business Stakeholdern und Mitglieder anderer Scrum Teams können ebenfalls in den War Room gehen und relevante Angelegenheiten besprechen.

10.2.2.4 Videokonferenzen

Im echten Leben ist es nicht immer möglich das gesamte Scrum Team an einem Ort unterzubringen. In diesen Fällen ist es notwendig, das Hilfsmittel der Videokonferenz zu benutzen, um direkte Kommunikation zu ermöglichen.

10.2.2.5 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

10.2.3 Outputs

10.2.3.1 Aktualisiertes Scrumboard*

Das Scrumboard wird weiterhin regelmäßig aktualisiert, wenn das Team Aufgaben erledigt. Weitere Informationen über das Scrumboard finden Sie in den Abschnitten 9.3.3.3 und 9.4.3.2.

10.2.3.2 Aktualisiertes Impediment Log*

Siehe Abschnitt 10.1.1.4.

10.2.3.3 Aktualisiertes Sprint Burndown oder Burnup Chart*

Das Sprint Burndown Chart sollte täglich aktualisiert werden, um den Fortschritt des Scrum Teams zu zeigen und auch um zu erkennen, ob die Abschätzungen möglicherweise falsch waren. Wenn das Sprint Burndown Chart zeigt, dass das Scrum Team nicht auf Kurs ist, um die Aufgaben im Sprint rechtzeitig abzuschließen, sollte der Scrum Master alle Hindernisse oder Hemmnisse für einen erfolgreichen Abschluss identifizieren und versuchen, diese zu beseitigen. Für weitere Informationen zum Sprint Burndown Chart siehe Abschnitt 9.6.3.3.

10.2.3.4 Motiviertes Scrum Team

Hinter dem Täglichen Standup Meeting steckt die Idee, dass jedes Teammitglied wichtig und ein unverzichtbarer Mitarbeiter ist. Das verbessert sowohl die Moral des Individuums als auch die des Teams. Zusammen mit dem Konzept des selbst organisierten Teams verbessert dies die generelle Motivation und führt zu effizienterer Teamleistung und besserer Qualität der zu liefernden Ergebnisse.

Das Scrum Team wird in Abschnitt 8.3.3.1 beschrieben.

10.2.3.5 Nicht genehmigte Change Requests

Siehe Abschnitt 8.4.1.6.

10.2.3.6 Identifizierte Risiken

Siehe Abschnitt 8.4.3.4.

10.2.3.7 Gemilderte Risiken

Siehe Abschnitt 10.1.3.6.

10.2.3.8 Aktualisierte Abhängigkeiten

Siehe Abschnitt 9.3.3.3.

10.3 Priorisiertes Produkt Backlog verfeinern

Das ist ein Prozess, in dem der Product Owner das Priorisierte Produkt Backlog ständig aktualisiert und pflegt. Ein Priorisiertes Produkt Backlog Review Meeting kann abgehalten werden. In ihm werden alle Veränderungen oder Updates des Backlog besprochen und entsprechend in das Priorisierte Produkt Backlog eingetragen.

Um das priorisierte Produkt Backlog bei Änderungen der Anforderungen und/oder Prioritäten auf dem neuesten Stand zu halten, arbeitet der Product Owner kontinuierlich mit dem Kunden und anderen Business Stakeholdern des Unternehmens zusammen, um alle Änderungen ihrer Bedürfnisse zu erfassen und zu verstehen.

Um sicherzustellen, dass die User Storys, für die der Product Owner vom Team im nächsten Sprint eine Zusage erhalten möchte, bereit sind, verfeinert der Product Owner die bestehenden Epics und User Storys im priorisierten Produkt Backlog und stellt sicher, dass die User Storys die Definition von Bereit erfüllen.

Als Teil der Verfeinerung des priorisierten Produkt Backlogs arbeitet der Product Owner auch mit dem Scrum Team zusammen, um Feedback und Fragen in Bezug auf die Aktualisierungen im priorisierten Produkt Backlog zu erhalten, möglicherweise einschließlich Abschätzungen.

Wenn Änderungen der Anforderungen und/oder des Gesamtfortschritts des Scrum-Teams Änderungen des Release-Plans und/oder der geschäftlichen Rechtfertigung erfordern, nimmt der Product Owner diese Änderungen ebenfalls während dieses Prozesses vor.

Dies ist der Prozess, in dem der Product Owner die meiste Zeit investieren wird.

Abbildung 10-8 zeigt Inputs, Hilfsmittel und Outputs für den *Tägliches Standup durchführen* Prozess.

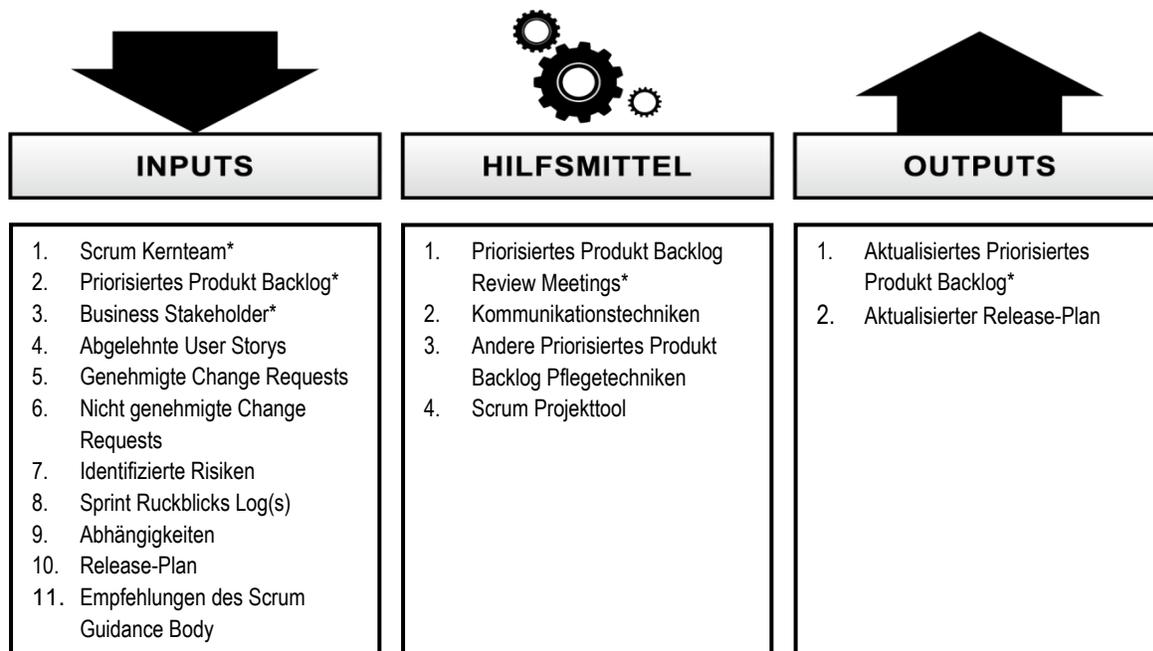


Abbildung 10-8: Priorisiertes Produkt Backlog verfeinern–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

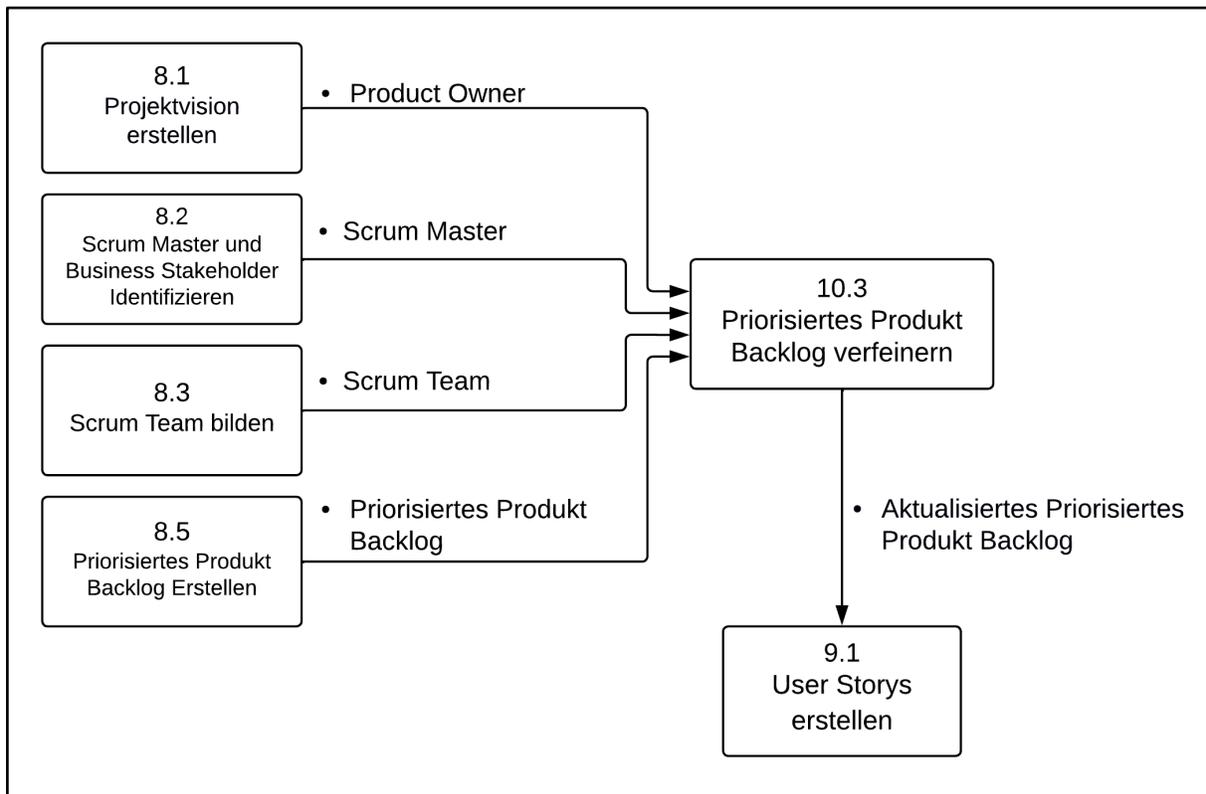


Abbildung 10-9: Priorisiertes Produkt Backlog verfeinern–Datenflussdiagramm

10.3.1 Inputs

10.3.1.1 Scrum Kernteam*

Siehe Abschnitts 8.1.3.1, 8.2.3.1 and 8.3.3.1.

10.3.1.2 Priorisiertes Produkt Backlog*

Siehe Abschnitt 8.5.3.1.

10.3.1.3 Business Stakeholdern*

Um das priorisierte Produkt Backlog im Hinblick auf geänderte Anforderungen und/oder Prioritäten auf dem neuesten Stand zu halten, arbeitet der Product Owner kontinuierlich mit dem Kunden und anderen Geschäftsinteressenten zusammen, um alle Änderungen ihrer Bedürfnisse zu erfassen und zu verstehen.

10.3.1.4 Abgelehnte User Storys

Wenn eine User Story nicht den Abnahmekriterien entspricht, wird diese als abgelehnte User Story betrachtet. Abgelehnte User Storys werden normalerweise nicht in einer separaten Liste festgehalten. Sie bleiben einfach im Priorisierten Produkt Backlog und werden nicht als erledigt markiert – so können sie im Prozess *Priorisiertes Produkt Backlog verfeinern* wieder priorisiert und für die weitere Entwicklung im nächsten oder einem anderen zukünftigen Sprint in Betracht gezogen werden.

10.3.1.5 Genehmigte Change Requests

Siehe Abschnitt 8.4.1.5.

10.3.1.6 Nicht genehmigte Change Requests

Siehe Abschnitt 8.4.1.6.

10.3.1.7 Identifizierte Risiken

Siehe Abschnitt 8.4.3.4.

10.3.1.8 Sprint Rückblick Log(s)

Siehe Abschnitt 11.3.3.4.

10.3.1.9 Abhängigkeiten

Siehe Abschnitt 9.3.3.3.

10.3.1.10 Release-Plan

Siehe Abschnitt 8.6.3.1.

10.3.1.11 Empfehlungen des Scrum Guidance Body

Während des Prozesses *Priorisiertes Produkt Backlog verfeinern* können die Empfehlungen des Scrum Guidance Body die besten Praktiken darüber enthalten, wie man systematisch Anforderungen von Business Stakeholdern und den Scrum Teams verstehen und sammeln kann – und wie man dann das Produkt Backlog angemessen priorisiert und allen relevanten Personen, die im Scrum Projekt involviert sind, die Aktualisierungen mitteilt. Mehr Informationen zu den Empfehlungen des Scrum Guidance Body finden Sie im Abschnitt 8.1.1.7

10.3.2 Hilfsmittel

10.3.2.1 Priorisiertes Produkt Backlog Review Meetings*

Der Product Owner kann mehrere und separate Meetings mit relevanten Business Stakeholdern, dem Scrum Master und dem Scrum Team abhalten. Somit kann er/sie sicherstellen, dass er/sie über ausreichende Informationen verfügt, um Aktualisierungen am Priorisiertem Produkt Backlog während des Prozesses *Priorisiertes Produkt Backlog verfeinern* durchzuführen.

Die Absicht des Priorisierten Produkt Backlog Review Meetings ist es, sicherzustellen, dass die User Storys und Abnahmekriterien verstanden und richtig vom Product Owner geschrieben worden sind, sodass sie die tatsächlichen Anforderungen und Prioritäten des Business Stakeholders (Kunden) reflektieren; dass User Storys von jedem im Scrum Team verstanden werden; dass User Storys mit hoher Priorität verfeinert werden, sodass das Scrum Team diese User Storys richtig abschätzen und umsetzen kann.

Das Priorisiertes Produkt Backlog Review Meeting stellt ebenfalls sicher, dass irrelevante User Storys entfernt werden und alle genehmigten Change Requests oder identifizieren Risiken in das Priorisierte Produkt Backlog eingetragen werden.

10.3.2.2 Kommunikationstechniken

Scrum begünstigt eine akkurate und effektive Kommunikation, und zwar grundsätzlich durch die Arbeit des Scrum Teams am gleichen Platz. Scrum bevorzugt informelle Interaktion von Angesicht zu Angesicht gegenüber geschriebener Kommunikation. Wenn es unumgänglich ist, dass das Team verteilt ist, muss der Scrum Master dafür sorgen, dass effektive Kommunikationstechniken vorhanden sind, sodass sich das Team selbst organisieren und effektiv arbeiten kann.

10.3.2.3 Andere Priorisiertes Produkt Backlog Verfeinerungstechniken

Einige andere Hilfsmittel zur Verfeinerung des priorisierten Produkt Backlogs sind die gleichen Hilfsmittel, die in den folgenden Prozessen genutzt wurden.

- *Epic(s) entwickeln* – Siehe Abschnitt 8.4.2.
- *Priorisiertes Produkt Backlog erstellen* – Siehe Abschnitt 8.5.2.
- *Release-Planung durchführen* – Siehe Abschnitt 8.6.2.
- *User Storys erstellen* – Siehe Abschnitt 9.1.2.
- *User Storys abschätzen* – Siehe Abschnitt 9.2.2.
- *User Storys Zusagen* Siehe Abschnitt 9.3.2.
- *Aufgaben Identifizieren* – Siehe Abschnitt 9.4.2.
- *Aufgaben abschätzen* – Siehe Abschnitt 9.5.2

10.3.2.4 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

10.3.3 Outputs

10.3.3.1 Aktualisiertes Priorisiertes Produkt Backlog*

Siehe Abschnitt 8.5.3.1

Das Priorisiertes Produkt Backlog kann mit neuen User Storys, neuen Change Requests, neu identifizierten Risiken, aktualisierten User Storys oder der Re-Priorisierung existierender User Storys, aktualisiert werden.

10.3.3.2 Aktualisierter Release-Plan

Siehe Abschnitt 8.6.3.1

Der Release-Plan kann aktualisiert werden, um die Auswirkung einer neuen oder geänderten User Story im Priorisierten Produkt Backlog zu reflektieren.

10.4 Datenflussdiagramm der Implementierungs-Phase

In Abbildung 10-10 ist das Datenflussdiagramm für die Implementierungsphase dargestellt.

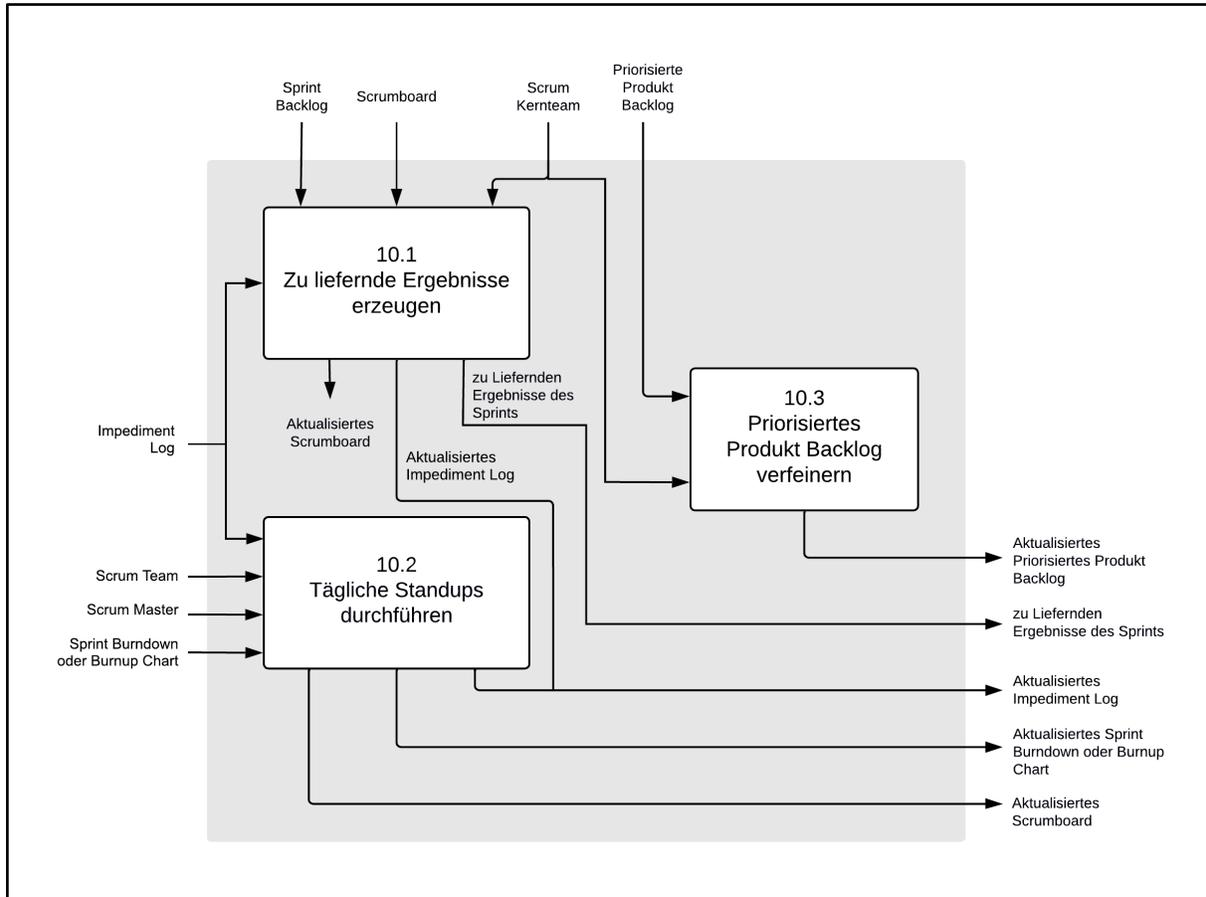


Abbildung 10-10: Implementierungsphase-Datenflussdiagramm

11. Review und Rückblick

Die Review- und Rückblicksphase beschäftigt sich mit der Überprüfung der zu liefernden Ergebnisse und der geleisteten Arbeit, sowie mit der Bestimmung von Möglichkeiten, die bei der Projektarbeit verwendeten Verfahren und Methoden zu verbessern. In großen Organisationen kann die Review- und Rückblicksphase auch den Scrum der Scrums beinhalten.

Review und Rückblick kann nach der Definition des Scrum Body of Knowledge (SBOK®) auf folgende Bereiche angewendet werden:

- Portfolios, Programme, und / oder Projekte in jeder Branche
- Produkte, Dienstleistungen oder andere Ergebnisse, die an die Business Stakeholdern geliefert werden
- Projekte von jeder Größe und Komplexität

Der Begriff "Produkt" bezieht sich im Rahmen dieses Body of Knowledge auf ein Produkt, eine Dienstleistung, oder ein anderes zu lieferndes Ergebnis. Scrum kann effektiv auf jedes Projekt in jeder Branche angewendet werden - von kleinen Projekten oder Teams mit eventuell nur sechs Teammitgliedern hin zu großen, komplexen Projekten mit bis zu mehreren hundert Teammitgliedern.

Um die beste Anwendung des Scrum Rahmens zu ermöglichen, kennzeichnet dieses Kapitel Inputs, Hilfsmittel und Outputs für jeden Prozess entweder als „verbindlich“ oder „optional“. Inputs, Hilfsmittel und Outputs, die mit einem Sternchen (*) gekennzeichnet sind, sind verbindlich oder werden als kritisch fuer den Erfolg des Projekts angesehen, wohingegen jene ohne Sternchen optional sind.

Es wird empfohlen, dass das Scrum Team und diejenigen, welche sich erst mit dem Scrum Rahmen vertraut machen, sich primär mit den verbindlichen Inputs, Hilfsmitteln und Outputs beschäftigen, während Product Owner, Scrum Master, und andere Scrum Anwender mit größerer Erfahrung danach streben sollten, sich ein noch tieferes Wissen über die Informationen in diesem gesamten Kapitel anzueignen. Darüber hinaus ist es ebenfalls von Bedeutung zu verstehen, dass, obwohl alle Prozesse einzeln im SBOK® definiert sind, diese nicht notwendigerweise auch nacheinander oder von einander getrennt ablaufen. Mitunter ist es nützlicher, einige Prozesse, je nach den speziellen Bedürfnissen eines jeden Projekts, miteinander zu kombinieren.

Dieses Kapitel ist aus der Perspektive eines einzelnen Scrum Teams geschrieben worden, welches an einem Sprint arbeitet, um potenziell versandfähige Ergebnisse als Teil eines größeren Projektes, eines Programms oder eines Portfolios zu produzieren. Weitere Informationen, die sich auf den Gebrauch von Scrum für große Projekte beziehen, sind in Kapitel 13 zu finden, die Skalierung von Scrum für Programme und Portfolios befinden sich in Kapitel 14.

Review und Rückblick ist die dritte und letzte der drei Phasen, die in jedem Sprint wiederholt werden.

Der Product Owner und die relevanten Business Stakeholdern überprüfen die vom Team erstellten zu Liefernden Ergebnisse und geben Feedback. Der Product Owner bewertet jede User Story und stellt fest, ob sie die jeweiligen Abnahmekriterien erfüllt und nimmt sie dementsprechend entweder an oder lehnt sie ab.

Im letzten Teil eines Sprints erarbeitet das Scrum-Team Möglichkeiten zur kontinuierlichen Verbesserung seiner Arbeit.

Es ist auch wichtig zu wissen, dass, obwohl alle Phasen und Prozesse im SBOK® Guide eindeutig definiert sind, sie nicht unbedingt sequenziell oder separat durchgeführt werden. Je nach den spezifischen Anforderungen des jeweiligen Projekts kann es manchmal sinnvoller sein, dass sich einige Phasen und/oder Prozesse überlappen.

Abbildung 11-1 gibt einen Überblick über die Prozesse der Review- und Rückblicksphase, welche nachstehend gelistet sind:

11.1 Sprint vorführen und validieren- In diesem Prozess zeigt das Scrum Team dem Product Owner die zu liefernden Ergebnisse des Sprints im Rahmen eines Sprint Review Meetings. Das Ziel dieses Meetings ist es, die Abnahme der User Stories des Sprints durch den Product Owner zu erlangen.

Dieser Prozess ist nicht nur ein wichtiges Qualitätselement in einem Scrum-Projekt, sondern auch ein Schlüsselement zur Aufrechterhaltung des Engagements der Business Stakeholdern. Die Business Stakeholdern sind ermuntert, am Sprint Review Meeting teilzunehmen, um sich aus erster Hand über das Produkt oder den Service und dessen Fortschritt zu informieren und Feedback zu geben. Das Feedback der Business Stakeholdern ist ein wichtiger Input für zukünftige Sprints.

11.2 Sprint Rückblick halten-In diesem Prozess treffen sich Scrum Master und Scrum Team, um die während des Sprints gemachten Erfahrungen zu besprechen. Diese Informationen werden als Lehren, die auf zukünftige Sprints angewendet werden können, dokumentiert. Oft kann es als Ergebnis dieser Diskussion zu Vereinbarten Verbesserungsmaßnahmen oder aktualisierten Empfehlungen des Scrum Guidance Body kommen. Dieser Prozess ist ein wichtiger Bestandteil der kontinuierlichen Verbesserung in Scrum.

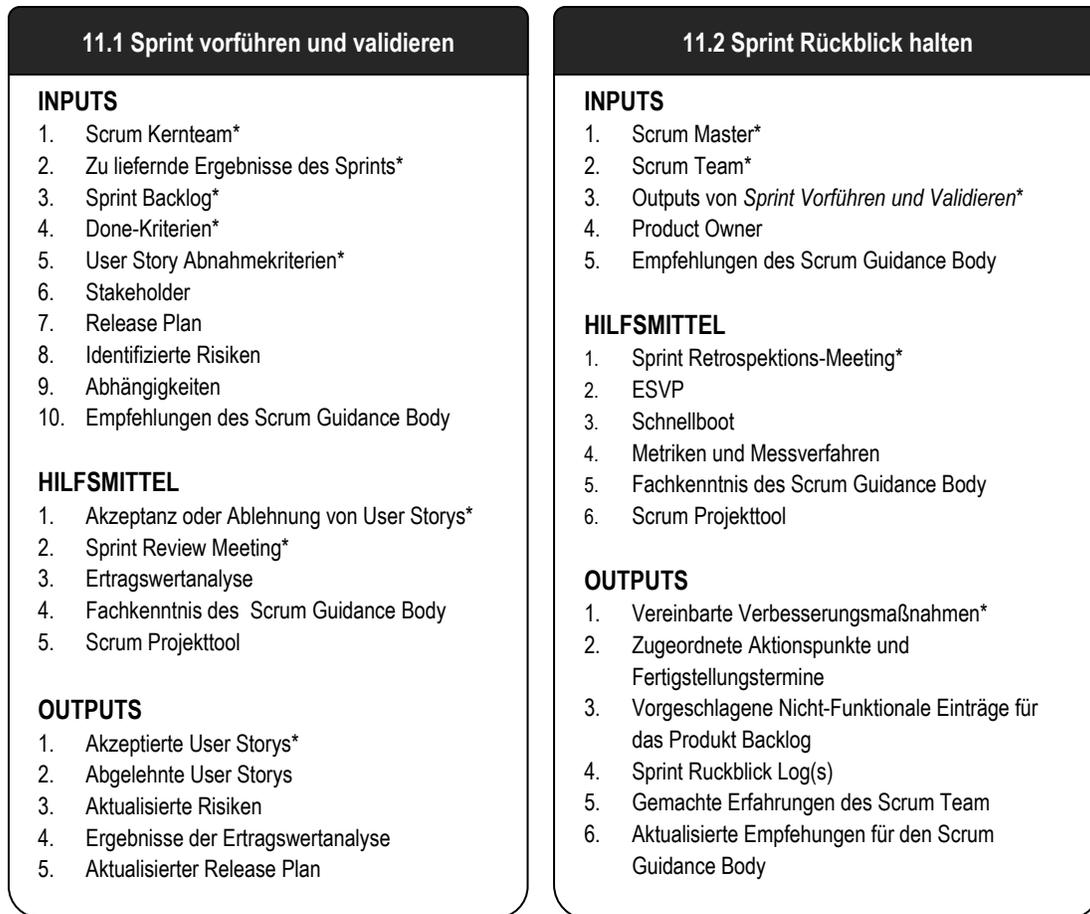


Abbildung 11-1: Überblick der Review- und Rückblicksphase

Abbildung 11-2 unten zeigt die verbindlichen Inputs, Hilfsmittel und Outputs für Prozesse in Review und Rückblicksphase.



Abbildung 11-2: Überblick der Review und Rückblicksphase (Kernpunkte)

11.1 Sprint Vorführen und Validieren

In diesem Prozess demonstriert das Scrum Team dem Product Owner die zu Liefernden Ergebnisse des Sprints in einem Sprint Review Meeting. Der Zweck dieses Meetings ist es, die Abnahme der User Storys des Sprints durch den Product Owner zu erhalten.

Dieser Prozess ist nicht nur ein wichtiges Qualitätselement in einem Scrum-Projekt, sondern auch ein zentrales Element zur Aufrechterhaltung des Engagements der Business Stakeholdern. Die Business Stakeholdern sind aufgefordert, am Sprint Review Meeting teilzunehmen, um sich aus erster Hand über das Produkt oder den Service und seinen Fortschritt zu informieren und Feedback zu geben. Das Business Stakeholdern-Feedback ist ein wichtiger Input für zukünftige Sprints.

Abbildung 11-3 zeigt Inputs, Hilfsmittel und Outputs für den *Sprint Vorführen und Validieren* Prozess.

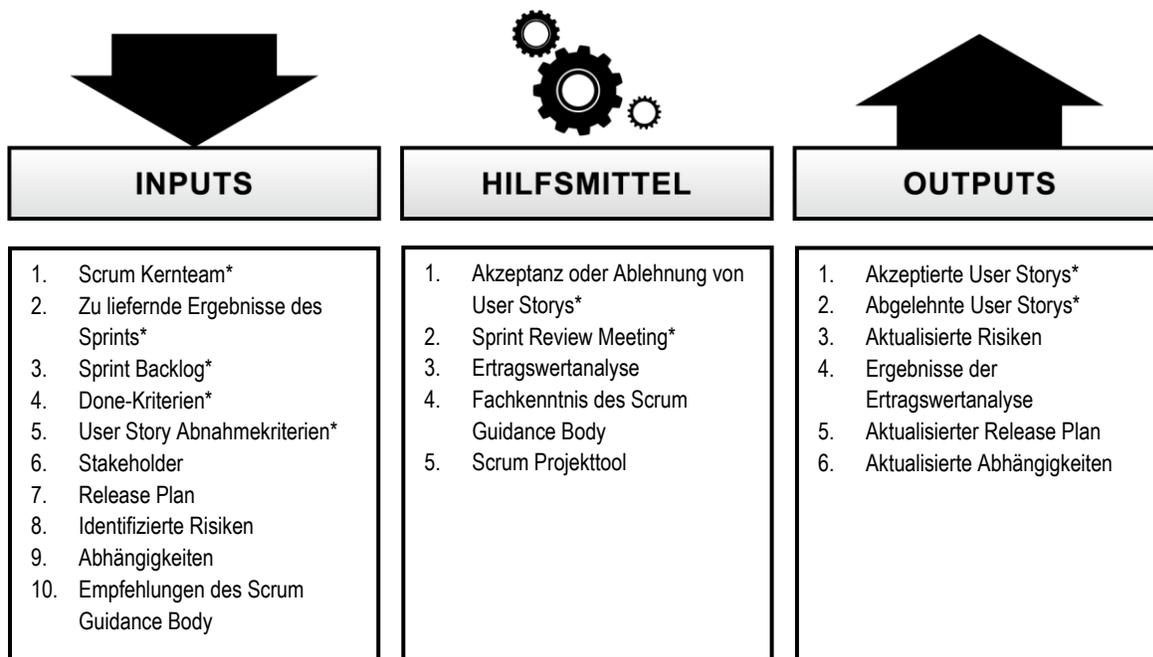


Abbildung 11-3: Sprint Vorführen und Validieren–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

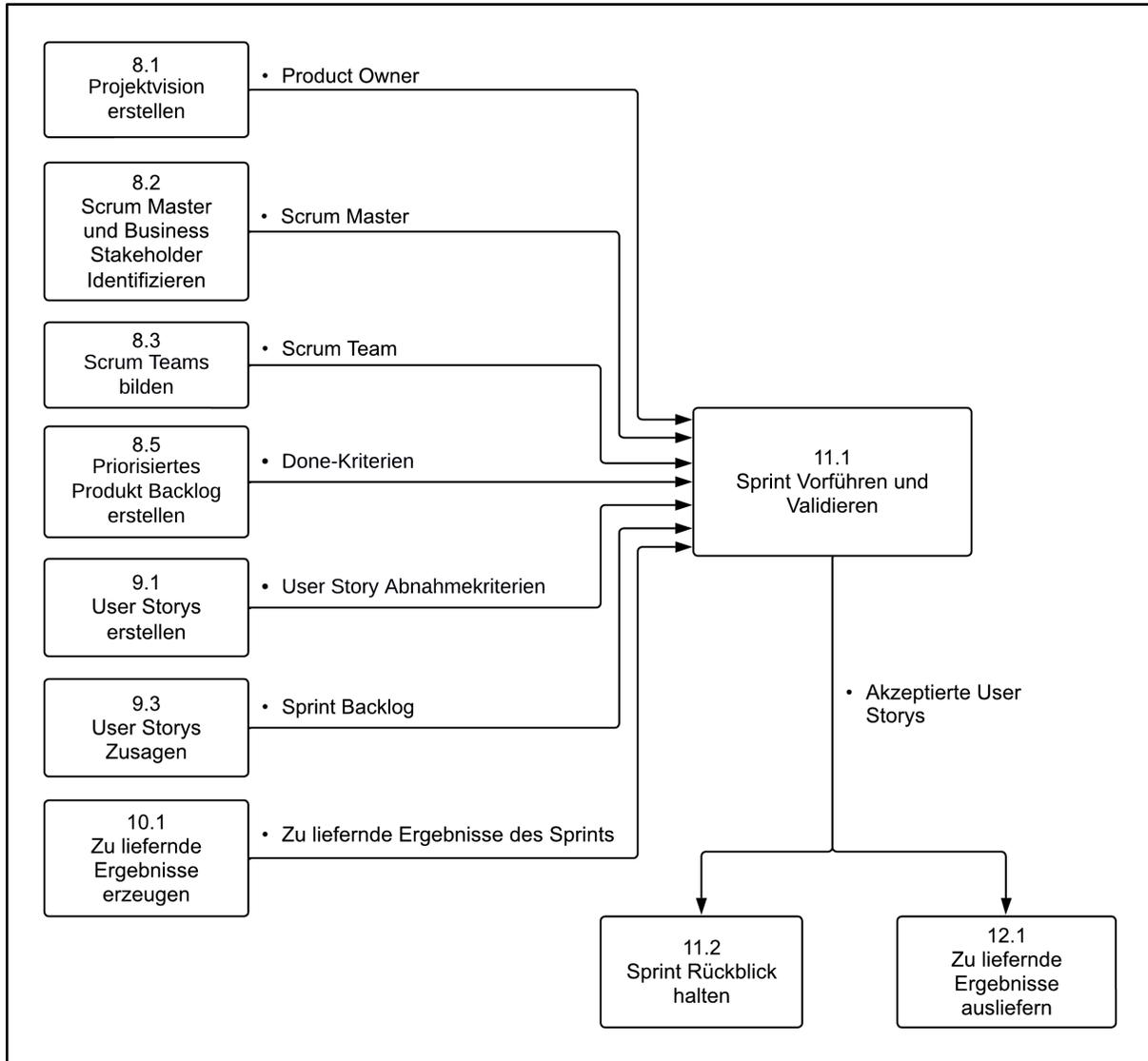


Abbildung 11-4: Sprint Vorführen und Validieren–Datenflussdiagramm

11.1.1 Inputs

11.1.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

11.1.1.2 Zu liefernde Ergebnisse des Sprints*

Siehe Abschnitt 10.1.3.1.

11.1.1.3 Sprint Backlog*

Siehe Abschnitt 9.5.3.1.

11.1.1.4 Done-Kriterien*

Siehe Abschnitt 8.5.3.2.

11.1.1.5 User Story Abnahmekriterien*

Siehe Abschnitt 9.4.1.3.

11.1.1.6 Business Stakeholdern

Siehe Abschnitt 8.2.3.2.

11.1.1.7 Release Plan

Siehe Abschnitt 8.6.3.1.

11.1.1.8 Identifizierte Risiken

Siehe Abschnitt 8.4.3.4.

11.1.1.9 Abhängigkeiten

Siehe Abschnitt 9.3.3.3

11.1.1.10 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11

Bei der Vorführung und Validierung des Sprints können Empfehlungen des Scrum Guidance Body die besten Verfahren beinhalten, um Sprint Review Meetings abzuhalten und die Ergebnisse der Ertragswertanalyse zu bewerten. Darüber hinaus können Richtlinien für das Teilen von Erfahrungen mit anderen Personen im Scrum Kernteam und anderen Scrum Teams im Projekt enthalten sein.

11.1.2 Hilfsmittel

11.1.2.1 Akzeptanz oder Ablehnung von User Storys*

Nachdem die User Storys vom Scrum Team fertiggestellt wurden, werden sie dem Product Owner zur Überprüfung zur Verfügung gestellt. Der Product Owner kann eine User Story überprüfen, sobald sie fertiggestellt ist, oder alle User Storys gemeinsam in einem Sprint Review Meeting am Ende des Sprints. Der Product Owner akzeptiert die User Storys, die die User Story Abnahmekriterien und die Done Kriterien erfüllen, und lehnt diejenigen ab, die die Kriterien nicht erfüllen, zusammen mit einem Feedback, warum eine User Story abgelehnt wurde. Wenn im aktuellen Sprint noch Zeit zur Verfügung steht, stehen abgelehnte User Stories dem Team zur Verfügung, um die Gründe für die Ablehnung zu beseitigen, und dem Product Owner im selben Sprint erneut zur Überprüfung vorzulegen. Am Ende eines Sprints verbleiben alle abgelehnten User Stories im priorisierten Product Backlog, um in einem zukünftigen Sprint zur Fertigstellung berücksichtigt zu werden.

11.1.2.2 Sprint Review Meeting*

Sprint Review Meetings werden am Ende eines jeden Sprints einberufen. Die Mitglieder des Scrum Kernteams und die relevanten Business Stakeholdern nehmen an den Sprint Review Meetings teil, um die zu Liefernden Ergebnisse zu präsentieren. Das Scrum Team demonstriert die im Rahmen des Sprints erstellten User Storys und zu Liefernden Ergebnisse, einschließlich der neu erstellten Funktionalitäten oder Produkte. User-Storys, die bereits vor dem Sprint-Review-Meeting vom Product Owner genehmigt wurden, werden vom Team während dieses Meetings ebenfalls vorgeführt, um sicherzustellen, dass der/die Business Stakeholdern die jeweilige Funktionalität ebenfalls begutachten. Der Product Owner und der/die Business Stakeholdern begutachten alle Ergebnisse und entscheiden, ob in einem nachfolgenden Sprint Änderungen vorgenommen werden müssen.

Am Ende des Sprint Review Meetings werden alle User Storys im Sprint in Betracht gezogen und einige vom Product Owner genehmigt und andere abgelehnt, je nachdem, ob sie die Abnahmekriterien und die Done Kriterien erfüllen.

11.1.2.3 Ertragswertanalyse

Siehe Abschnitt 4.6.1

11.1.2.4 Fachkenntnis des Scrum Guidance Body

Siehe Abschnitt 8.4.2.7.

Bei der Vorführung und Validierung des Sprints kann sich Fachkenntnis des Scrum Guidance Body auf die besten Methoden für die Durchführung eines Sprint Retrospektions-Meetings beziehen. Es kann darüber hinaus Experten geben, welche bei der Bereitstellung von Richtlinien zur besseren Leitung von Sprint Retrospektions-Meetings helfen können.

11.1.2.5 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

11.1.3 Outputs

11.1.3.1 Akzeptierte User Storys*

Das Ziel eines Sprints ist es, potenziell versandfähige zu liefernde Ergebnisse (oder Produkt-Inkrementen) zu erstellen, die die User Story Abnahmekriterien erfüllen. User Storys, die die Abnahmekriterien erfüllen, werden vom Product Owner formal akzeptiert. Diese akzeptierten User Storys können an die Kunden weitergegeben werden, wenn diese es wünschen. Es wird eine Liste der akzeptierten zu liefernden Ergebnisse geführt und nach jedem Sprint Review Meeting aktualisiert.

11.1.3.2 Abgelehnte User Storys

Wenn User Storys die Abnahmekriterien nicht erfüllen, gelten sie als unvollständig und werden vom Product Owner abgelehnt. Abgelehnte User Storys werden wieder in das priorisierte Product Backlog aufgenommen, so dass sie als Teil eines nachfolgenden Sprints berücksichtigt werden können.

Die Arbeit an den zu Liefernden Ergebnissen, die mit den abgelehnten User Storys in Zusammenhang stehen, kann von jedem Scrum Team erledigt werden, dem diese User Storys in Zukunft zugewiesen werden.

Da einige Arbeiten zur Erstellung der zu Liefernden Ergebnisse dieser abgelehnten User Storys bereits erledigt sein könnten, kann die zukünftige Abschätzung dieser User Storys, wenn sie erneut zur Bearbeitung in einem zukünftigen Sprint zugewiesen werden, geringer ausfallen als die ursprüngliche Abschätzung. In einigen Fällen können Scrum-Teams jedoch beschließen, die mit den abgelehnten User Storys verbundenen Ergebnisse vollständig zu ignorieren und die anstehenden Arbeiten als neue User Storys zu betrachten.

11.1.3.3 Aktualisierte Risiken

Siehe Abschnitt 8.4.3.4.

11.1.3.4 Ergebnisse der Ertragswertanalyse

Siehe Abschnitt 4.6.1.

11.1.3.5 Aktualisierter Release Plan

Siehe Abschnitt 10.3.3.2.

11.2 Sprint Rückblick halten

In diesem Prozess treffen sich Scrum Master und Scrum Team, um die während des Sprints gemachten Erfahrungen zu besprechen. Diese Informationen werden als Lehren, die auf zukünftige Sprints angewendet werden können, dokumentiert. Oft kann es als Ergebnis dieser Diskussion zu Vereinbarten Verbesserungsmaßnahmen oder aktualisierten Empfehlungen des Scrum Guidance Body kommen. Dieser Prozess ist ein wichtiger Bestandteil der kontinuierlichen Verbesserung in Scrum.

Abbildung 11-5 zeigt Inputs, Hilfsmittel und Outputs für den *Sprint Rückblick halten* Prozess.

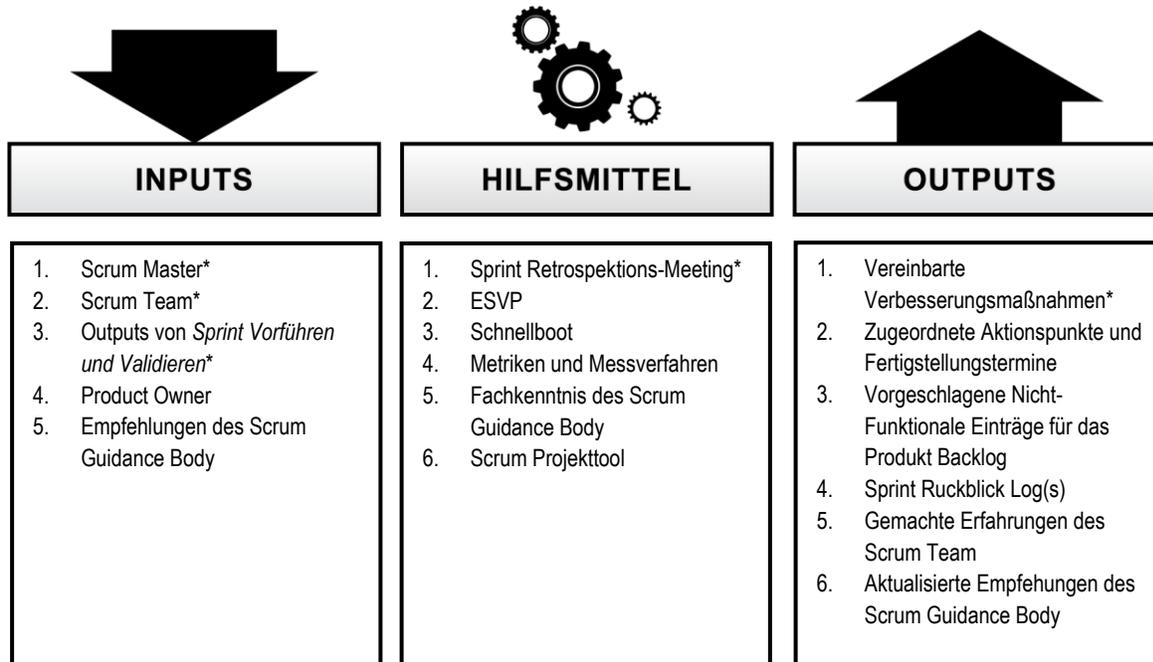


Abbildung 11-5: Sprint Rückblick halten–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

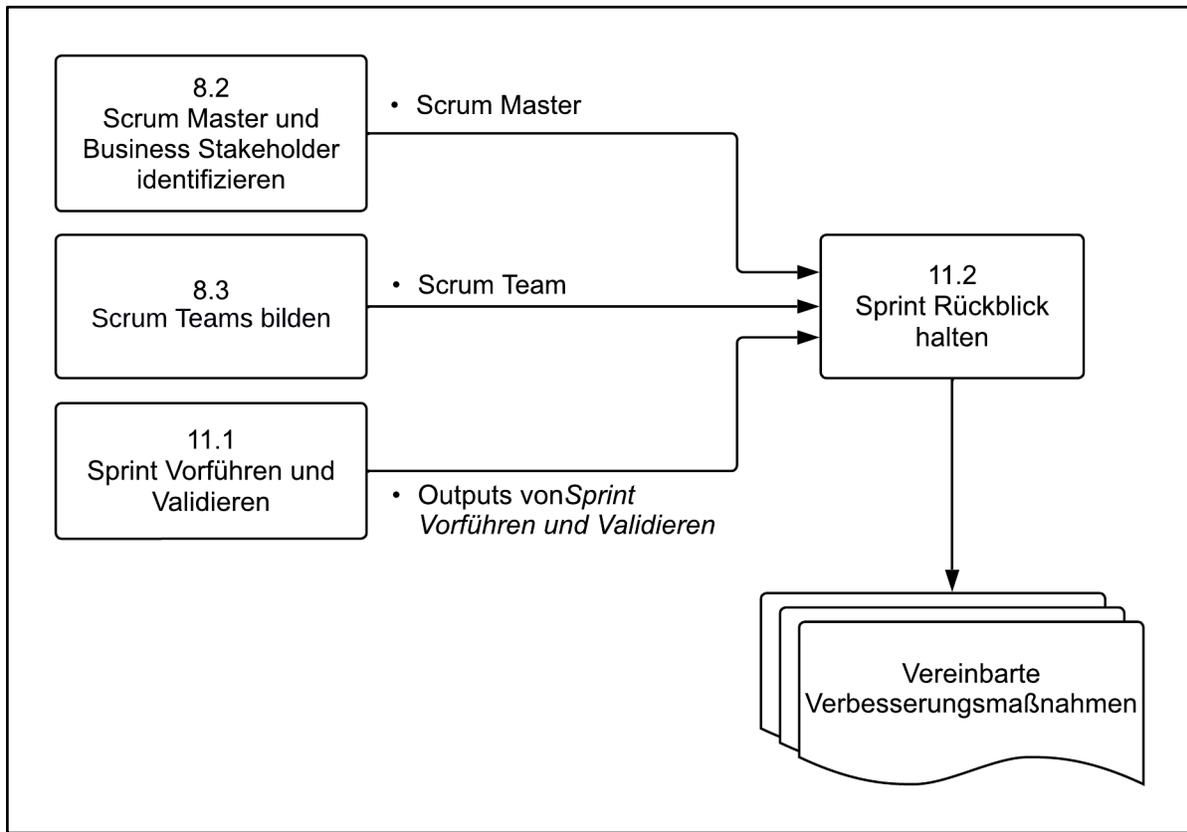


Abbildung 11-6: Sprint Rückblick halten– Datenflussdiagramm

11.2.1 Inputs

11.2.1.1 Scrum Master*

Siehe Abschnitt 8.2.3.1.

11.2.1.2 Scrum Team*

Siehe Abschnitt 8.3.3.1.

11.2.1.3 Outputs aus Sprint vorführen und validieren*

Siehe Abschnitt 11.2.3.

Die Ergebnisse aus *Sprint vorführen und validieren* bieten wertvolle Einblicke für den Prozess *Sprint Rückblick halten*.

11.2.1.4 Product Owner

Siehe Abschnitt 8.1.3.1.

11.2.1.5 Empfehlungen des Scrum Guidance Body

Der Scrum Guidance Body kann Richtlinien für die Durchführung des Sprint Retrospektions-Meetings, einschließlich von Empfehlungen für nutzbare Hilfsmittel und von den Meetings erwartete Dokumentationen und zu liefernde Ergebnisse, beinhalten.

11.2.2 Hilfsmittel

11.2.2.1 Sprint Retrospektions-Meeting*

Das Sprint Retrospektions-Meeting ist ein wichtiges Element des Scrum Konzeptes der Überprüfung und Anpassung und ist der letzte Schritt in einem Sprint. Alle Scrum Team-Mitglieder nehmen an der Sitzung teil, die vom Scrum Master geleitet oder moderiert wird. Es wird empfohlen, dass der Product Owner an der Sitzung teilnimmt, jedoch ist dies nicht verbindlich. Ein Teammitglied fungiert als Schriftführer und dokumentiert die Diskussionen und Punkte für das weitere Vorgehen.

Es ist wichtig, dieses Treffen in einer offenen und entspannten Atmosphäre abzuhalten, um eine uneingeschränkte Beteiligung aller Teammitglieder zu fördern. Die Diskussionen im Sprint Retrospektions-Meeting umfassen sowohl, was schlecht lief, als auch, was gut gelungen ist. Primäres Ziel des Treffens ist die Identifikation von drei spezifische Dingen:

- 1) Dinge, die das Team fortsetzen muss: die Besten Praktiken
- 2) Dinge, die das Team anfangen muss zu tun: Verbesserungen des Prozesses
- 3) Dinge, die das Team nicht weiter fortsetzen darf: Prozessprobleme und Engpässe

Diese Bereiche werden diskutiert und eine Liste von Vereinbarten Verbesserungsmaßnahmen wird erstellt.

11.2.2.2 Forscher–Käufer–Urlauber–Gefangener (ESVP)

Dies ist eine Übung, die zu Beginn des Sprint Retrospektions-Meetings durchgeführt werden kann, um die Denkweise der Teilnehmer zu verstehen und den Ton für die Sitzung festzulegen. Die Teilnehmer werden gebeten, anonym anzugeben, welche Rolle sie nach ihrem Gefühl im Meeting am besten repräsentiert.

- Forscher—Möchte an Allem teilnehmen und Alles erlernen, das innerhalb des Retrospektions-Meetings diskutiert wird
- Käufer—Möchte alles anhören und auswählen, was er aus dem Retrospektions-Meeting mitnimmt
- Urlauber—Möchte sich entspannen und als Tourist das Retrospektions-Meeting angehen
- Gefangener—Möchte an einem anderen Ort sein und nimmt am Retrospektions-Meeting teil, weil er muss

Der Scrum Master sammelt dann die Antworten, bereitet die Informationen auf und teilt sie mit der Gruppe.

11.2.2.3 Schnellboot

Schnellboot ist eine Technik, die verwendet werden kann, um das Sprint Retrospektions-Meeting durchzuführen. Die Team-Mitglieder spielen die Rolle der Crew auf einem Schnellboot. Das Boot muss eine Insel, die symbolisch für die Projektvision steht, erreichen. Haftnotizen werden von den Teilnehmern verwendet, um Antriebe und Anker darzustellen. Die Antriebe sind Dinge, die ihnen helfen, die Insel zu erreichen, während die Anker die Dinge darstellen, die sie vom Erreichen der Insel abhalten. Diese Übung ist zeitlich auf einige wenige Minuten begrenzt. Sobald alle Punkte dokumentiert sind, werden die Informationen zusammengetragen, diskutiert und über einen Abstimmungsprozess priorisiert. Antriebe werden anerkannt und für Anker werden entsprechend ihrer Priorität Minderungsmaßnahmen geplant.

11.2.2.4 Metriken und Messverfahren

Eine Vielzahl von Metriken kann verwendet werden, um die Leistung des Teams im gegenwärtigen Sprint zu messen und mit vorherigen Sprints zu vergleichen. Einige dieser Metriken sind:

- Teamgeschwindigkeit—Anzahl an absolvierten Story Points im jeweiligen Sprint
- Done Erfolgsrate—Prozentzahl der tatsächlich erledigten (done) Story Points im Vergleich zur Zahl der zugesagten Story Points.
- Effektivität der Abschätzungen—Zahl oder Prozentzahl von Abweichungen zwischen der geschätzten und der tatsächlichen Zeit, die für die Aufgaben und User Storys aufgewendet wurde
- Überprüfung von Feedback-Bewertungen—Feedback kann von den Business Stakeholdern unter Nutzung von quantitativen oder qualitativen Bewertungsmaßstäben erbeten werden, um einen Maßstab für die Leistung des Teams zu erhalten.
- Bewertung der Teammoral—Ergebnisse der Selbsteinschätzungen der Mitglieder zu Teammoral

- Peer-Feedback—360 Grad Feedback Mechanismen können genutzt werden, um konstruktive Kritik einzufordern und Einblick in die Leistung des Teams zu erhalten.
- Fortschritte zum Release—Bei jedem Release bereitgestellter wirtschaftlicher Wert, sowie der Wert der aktuellen Entwicklung auf dem Weg zu einem Release. Dies trägt zur Motivation des Teams und zur Erhöhung des Niveaus der Arbeitszufriedenheit bei.

11.2.2.5 Fachkenntnis des Scrum Guidance Body

Siehe Abschnitt 8.4.2.7.

Bei der Durchführung von Sprint Rückblicken können Fachkenntnisse des Scrum Guidance Body sich auf die besten Praktiken, um ein Sprint Retrospektions-Meeting durchzuführen, beziehen. Es kann darüber hinaus Experten geben, welche eine Anleitung zur Anwendung von Hilfsmitteln im Sprint Rückblick geben, um vereinbarte Verbesserungsmaßnahmen für zukünftige Sprints zu liefern.

11.2.2.6 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

11.2.3 Outputs

11.2.3.1 Vereinbarte Verbesserungsmaßnahmen*

Vereinbarte Verbesserungsmaßnahmen sind der primäre Output des Prozesses Sprint Rückblick halten. Sie sind die Liste der zu erledigenden Dinge, die das Team formuliert hat, um Probleme zu behandeln und Prozesse zu verbessern, um die Leistung des Teams bei zukünftigen Sprints zu verbessern.

11.2.3.2 Zugeordnete Aktionspunkte und Fertigstellungstermine

Sobald die Vereinbarten Verbesserungsmaßnahmen ausgearbeitet und verfeinert wurden, können Aktionspunkte zur Implementierung von Verbesserungen durch das Scrum Team in Betracht gezogen werden. Jedem Aktionspunkt wird ein Fertigstellungstermin zugewiesen.

11.2.3.3 Vorgeschlagene Nicht-Funktionale Einträge für das Produkt Backlog

Wenn das anfängliche Priorisierte Produkt Backlog entwickelt wird, basiert es auf User Storys und den erforderlichen Funktionalitäten. Oft werden nicht-funktionale Anforderungen nicht vollständig in den frühen Stadien des Projekts definiert und können im Sprint Review oder im Sprint Retrospektions-Meeting in Erscheinung treten. Diese Punkte sollten zum Priorisierten Produkt Backlog hinzugefügt werden, sobald sie entdeckt werden. Einige Beispiele für nicht-funktionale Anforderungen sind Reaktionszeiten, Kapazitätsgrenzen, und sicherheitsbezogene Angelegenheiten.

11.2.3.4 Sprint Rückblick Log(s)

Das Sprint Rückblick Log ist eine Aufzeichnung von den Meinungen, Diskussionen und Aktionspunkten, die bei einem Sprint Retrospektions-Meeting zur Sprache gekommen sind. Der Scrum Master kann die Erstellung dieses Protokolls anhand von Vorgaben des Scrum Kernteams durchführen. Die Sammlung aller Sprint Rückblick Logs bildet das Projekttagbuch und vermittelt die Erfolge, Probleme und Entscheidungen innerhalb des Projekts. Die Protokolle sind öffentliche Dokumente und für jeden in der Organisation zugänglich.

11.2.3.5 Gemachte Erfahrungen (Lessons Learned) des Scrum Teams

Von einem selbstorganisierenden und ermächtigten Scrum-Team wird erwartet, das es aus allen Fehlern, die während eines Sprint gemacht wurden, lernt - und diese gemachten Erfahrungen helfen den Teams ihre Leistung bei zukünftigen Sprints zu verbessern. Diese Lehren können auch in den Empfehlungen des Scrum Guidance Body dokumentiert werden, um diese anderen Teams zugänglich zu machen.

Es mag mehrere positive Lehren geben, die aus einem Sprint gezogen werden. Diese positiven Lehren sind ein wesentlicher Bestandteil des Sprint Rückblicks und sollten in angemessener Weise sowohl innerhalb des Teams als auch dem Scrum Guidance Body zugänglich gemacht werden, da das Team nach kontinuierlicher Selbst-Verbesserung strebt.

11.2.3.6 Aktualisierte Empfehlungen des Scrum Guidance Body

Als Ergebnis eines Sprint Retrospektions-Meetings können Vorschläge gemacht werden, um die Empfehlungen des Scrum Guidance Body zu überarbeiten oder zu verbessern. Wenn der Scrum Guidance Body diese Vorschläge annimmt, werden diese als Aktualisierungen in die Scrum Guidance Body-Dokumentation aufgenommen.

11.3 Datenflussdiagramm der Review- und Rückblicksphase

In Abbildung 11-7 ist das Datenflussdiagramm für die Review- und Rückblicksphase dargestellt.

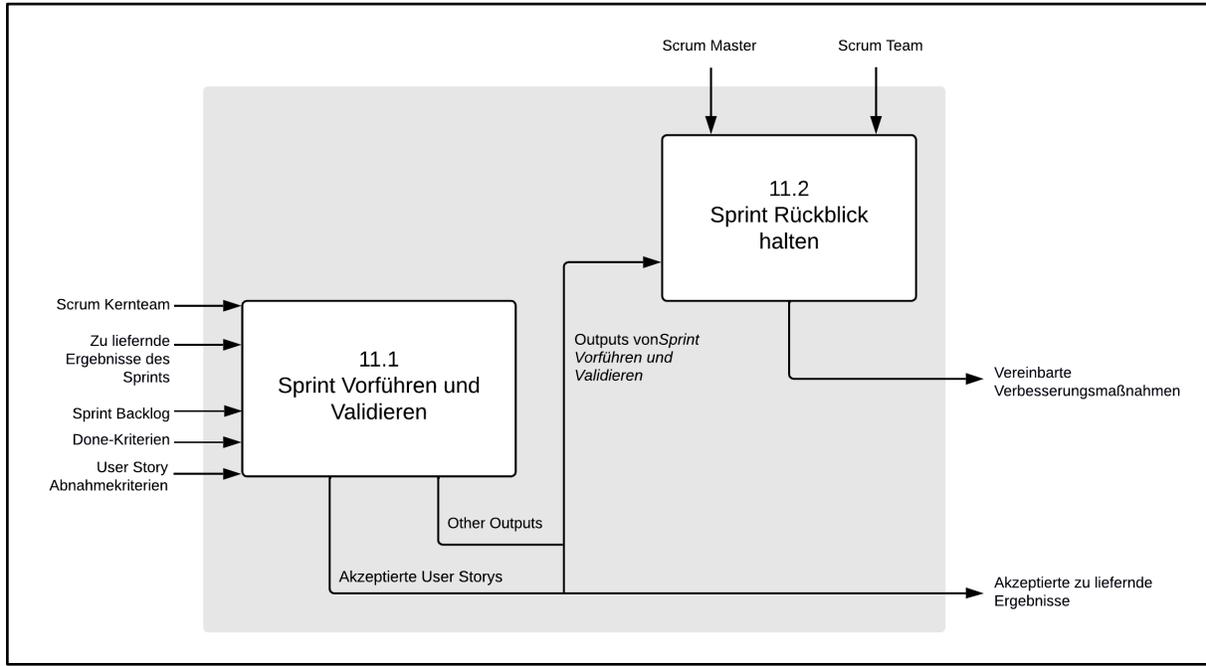


Abbildung 11-7: Übersicht und Rückblicksphase-Datenflussdiagramm

12. RELEASE

Die Releasephase hebt das Liefern der Akzeptierten zu liefernden Ergebnisse an den Kunden hervor. Während des Projekts gezogene Lehren werden identifiziert, dokumentiert und verinnerlicht.

Release, wie im *Scrum Body of Knowledge (SBOK®)* definiert, ist auf Folgendes anwendbar:

- Portfolios, Programme, und/oder Projekte in *jeder* Branche
- Produkte, Dienstleistungen, oder jegliche an Business Stakeholdern zu liefernde Ergebnisse
- Projekte jeglicher Größe oder Komplexität

Der Begriff "Produkt" in dieser Wissenssammlung kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann auf jedes Projekt in jeder Branche effektiv angewendet werden — von kleinen Projekten oder Teams mit lediglich sechs Team-Mitgliedern bis hin zu großen, komplexen Projekten mit mehreren Hundert Team-Mitgliedern.

Um die beste Anwendung des Scrum Rahmens zu ermöglichen, kennzeichnet dieses Kapitel Inputs, Hilfsmittel und Outputs für jeden Prozess entweder als „verbindlich“ oder „optional“. Inputs, Hilfsmittel und Outputs, die mit einem Sternchen (*) gekennzeichnet sind, sind verbindlich oder werden als kritisch fuer den Erfolg des Projekts angesehen, wohingegen jene ohne Sternchen optional sind.

Es wird empfohlen, dass das Scrum Team und alle Personen, die in den Scrum Rahmen und Prozesse eingeführt werden, ihren Fokus vorwiegend auf die verbindlichen Inputs, Hilfsmittel und Outputs richten. Product Owner, Scrum Master und weitere fortgeschrittene Scrum Praktiker werden ein genaueres Verständnis des gesamten Kapitels erreichen wollen. Es ist außerdem wichtig, sich bewusst zu machen, dass obwohl alle Prozesse einzeln im *SBOK®* definiert sind, sie nicht unbedingt sequentiell oder getrennt angewendet werden. Manchmal empfiehlt es sich, je nach spezifischen Bedürfnissen eines jeden Projektes, einige Prozesse miteinander zu kombinieren. Dieses Kapitel ist aus der Perspektive eines einzelnen Scrum Teams geschrieben worden, welches an einem Sprint arbeitet, um potenziell versandfähige Ergebnisse als Teil eines größeren Projektes, eines Programms oder eines Portfolios zu produzieren. Weitere Informationen, die sich auf den Gebrauch von Scrum für große Projekte beziehen, sind in Kapitel 13 zu finden, die Skalierung von Scrum für Programme und Portfolios befinden sich in Kapitel 14.

Die Release-Phase wird in der Regel mehrmals während eines Scrum-Projekts durchgeführt. Obwohl das Ergebnis jedes Sprints ein potenziell versandfähiges Produkt ist, gibt es nicht unbedingt nach jedem Sprint ein Release.

Zu Liefernde Ergebnisse aus akzeptierten User Stories eines oder mehrerer vorangegangener Sprints werden an relevante Business Stakeholdern zur Abnahme und Verwendung gemäß dem Release-Plan freigegeben.

Zusätzlich zur Freigabe von zu Liefernden Ergebnissen legen das Scrum Kernteam und die betrieblichen Business Stakeholdern fest, wie die Durchführung zukünftiger Releases im Projekt verbessert werden kann.

Es ist auch wichtig zu wissen, dass, obwohl alle Phasen und Prozesse im SBOK® Guide eindeutig definiert sind, sie nicht unbedingt sequenziell oder separat durchgeführt werden. Je nach den spezifischen Anforderungen des jeweiligen Projekts kann es manchmal sinnvoller sein, dass sich einige Phasen und/oder Prozesse überlappen.

Abbildung 12-1 gibt einen Überblick über die Prozesse der Release Phase, welche nachstehend gelistet sind:

12.1 Zu liefernde Ergebnisse ausliefern—In diesem Prozess werden Akzeptierte User Storys aus vorher abgeschlossenen Sprints ausgeliefert oder an die zuständigen Business Stakeholdern übergeben. Eine formale Vereinbarung über Funktionierende zu liefernde Ergebnisse dokumentiert den erfolgreichen Abschluss des Sprints.

12.2 Release Rückblick halten—In diesem Prozess, der ein Release abschließt, kommen Business Stakeholdern aus der Organisation und die Mitglieder des Scrum Kernteams zusammen, um auf das Projekt zurückzublicken und Lehren aus dem Projekt zu identifizieren, zu dokumentieren und zu verinnerlichen. Oft führen diese Lehren zur Dokumentation von vereinbarten Maßnahmen, die in zukünftigen Projekten umgesetzt werden.

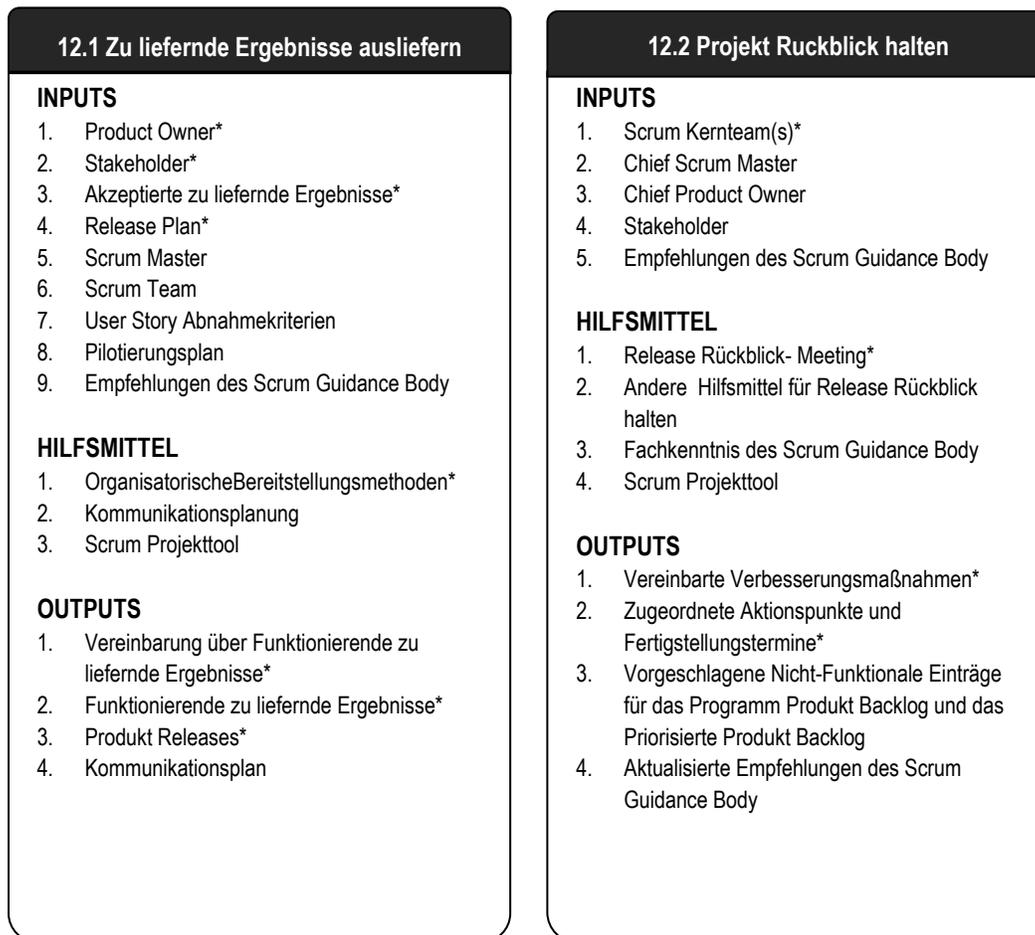


Abbildung 12-1: Release Übersicht

Abbildung 12-2 unten zeigt die verbindlichen Inputs, Hilfsmittel und Outputs für *Prozesse* in Releasephase.

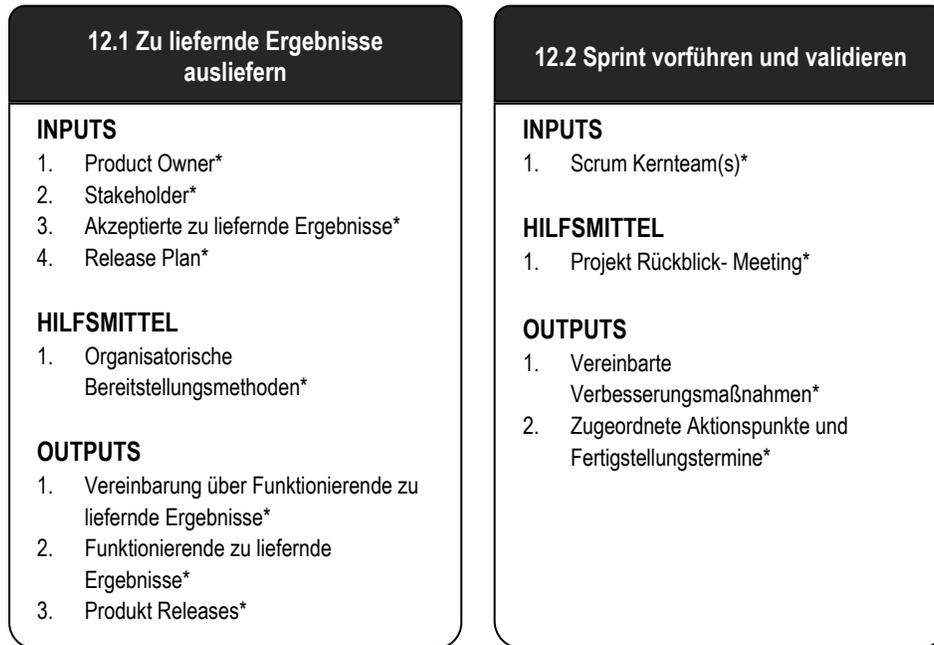


Abbildung 12-2: Release Übersicht (Kernpunkte)

12.1 Zu liefernde Ergebnisse ausliefern

In diesem Prozess werden Akzeptierte User Storys aus vorher abgeschlossenen Sprints ausgeliefert oder an die zuständigen Business Stakeholdern übergeben. Eine formale Vereinbarung über Funktionierende zu liefernde Ergebnisse dokumentiert den erfolgreichen Abschluss des Sprints.

Abbildung 12-3 zeigt Inputs, Hilfsmittel und Outputs für den *Zu liefernde Ergebnisse ausliefern* Prozess.

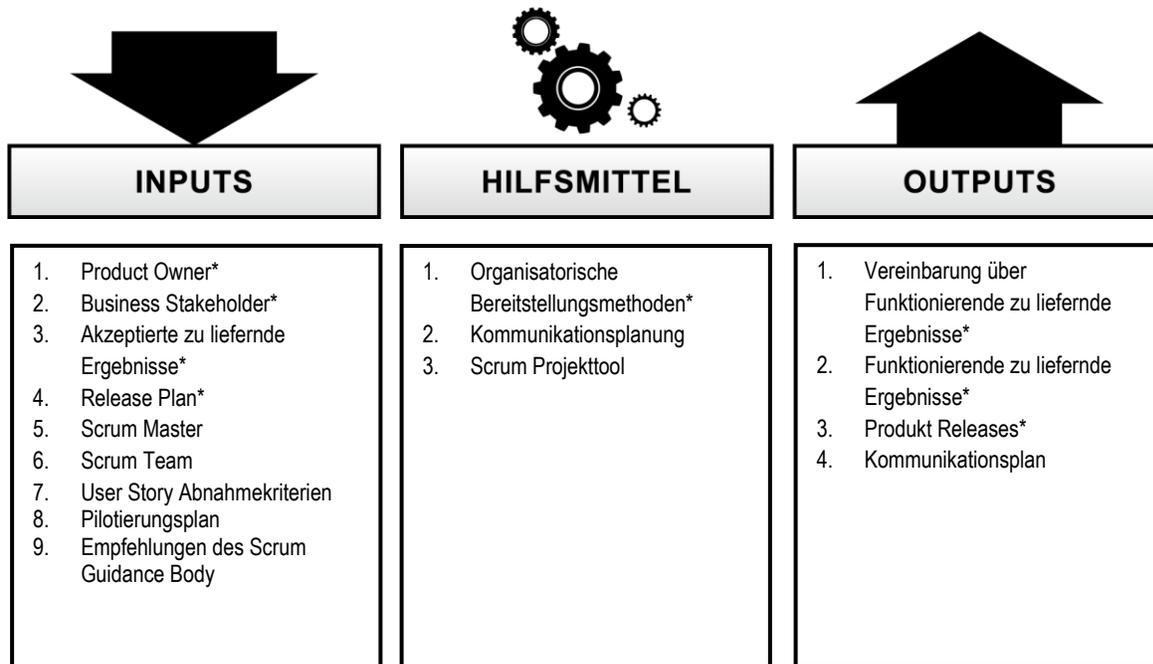


Abbildung 12-3: Zu liefernde Ergebnisse ausliefern–Inputs Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

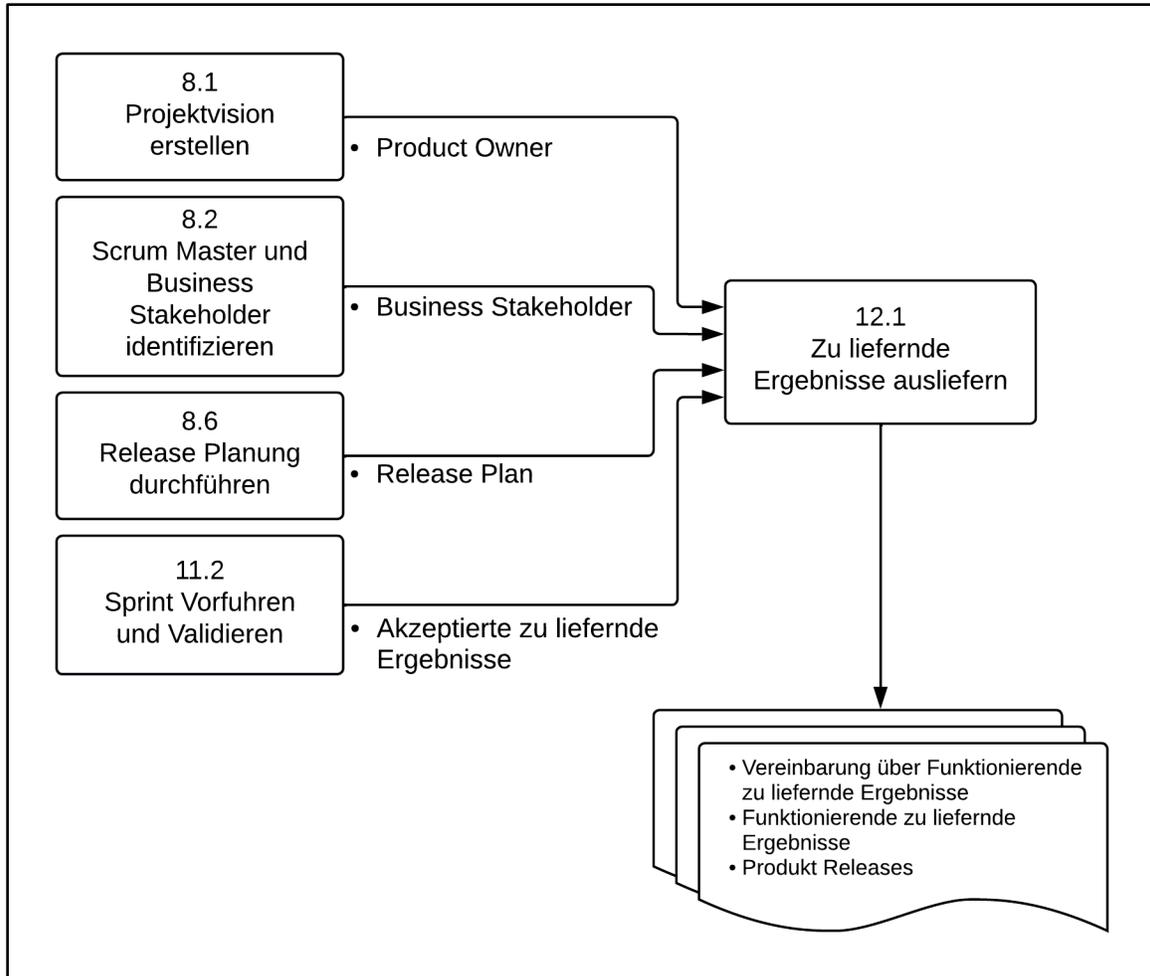


Abbildung 12-4: Zu liefernde Ergebnisse ausliefern – Datenflussdiagramm

12.1.1 Inputs

12.1.1.1 Product Owner*

Siehe Abschnitt 8.1.3.1.

12.1.1.2 Business Stakeholdern*

Siehe Abschnitt 8.2.3.2.

12.1.1.3 Akzeptierte zu liefernde Ergebnisse*

Siehe Abschnitt 11.2.3.1.

12.1.1.4 Release Plan

Siehe Abschnitt 8.6.3.1.

12.1.1.5 Scrum Master

Siehe Abschnitt 8.2.3.1.

12.1.1.6 Scrum Team

Siehe Abschnitt 8.3.3.1.

12.1.1.7 User Story Abnahmekriterien

Siehe Abschnitt 9.1.3.2.

12.1.1.8 Pilotierungsplan

Ein Pilotierungsplan kann verwendet werden, um eine Pilot-Bereitstellung im Detail auszuarbeiten. Umfang und Ziele der Bereitstellung, Ziel-Anwenderbasis, Bereitstellungszeitplan, Übergangspläne, benötigte Anwendervorbereitung, Bewertungskriterien für die Bereitstellung und andere wichtige Elemente, die im Zusammenhang mit der Bereitstellung stehen, werden im Pilotierungsplan spezifiziert und den Business Stakeholdern mitgeteilt.

12.1.1.9 Empfehlungen des Scrum Guidance Body

Beschreibung in Abschnitt 8.1.1.11

Während des *Zu liefernde Ergebnisse ausliefern* Prozesses kann der Empfehlungen des Scrum Guidance Body und Richtlinien enthalten, welche sich auf die Bereitstellung des Produktes beziehen. Es sind ausgezeichnete Praktiken, welche in Betracht gezogen werden sollten, wenn ein Produkt für den Kunden bereitgestellt werden soll, um den gelieferten Wert zu maximieren.

12.1.2 Hilfsmittel

12.1.2.1 Organisatorische Bereitstellungsmethoden*

Die Bereitstellungsmechanismen verschiedener Organisationen neigen dazu, abhängig von Industrie, Zielerwartungen und Positionierung unterschiedlich zu sein. Abhängig vom gelieferten Produkt kann die Bereitstellung aus der Ferne erfolgen, aber auch den physischen Versand oder die physische Übergabe eines Artikels beinhalten. Weil die Bereitstellung dahin tendiert ein hohes Risiko mit sich zu bringen, haben Organisationen normalerweise gut entwickelte und definierte Bereitstellungsmechanismen, welche detaillierte Prozesse beinhalten, um die Einhaltung von allen anwendbaren Standards und der Qualitätssicherungsmaßnahmen zu sicherzustellen. Diese können das Gegenzeichnen von bestimmten Repräsentanten des Managements, Anwender Abnahmemechanismen und Richtlinien, bezogen auf die minimale Funktionalität für das Release, enthalten.

12

12.1.2.2 Kommunikationsplan

In vielen Projekten gibt es einen Kommunikationsplan. Dieser Plan spezifiziert die während des Projekts anzulegenden und zu pflegenden Datensätze. Dabei wird eine Vielzahl an Methoden zur Vermittlung wichtiger Projektinformationen an die Business Stakeholdern angewendet. Der Kommunikationsplan definiert sowohl diese Methoden als auch die für diverse Kommunikationsaktivitäten Verantwortlichen. Wenn die zu liefernden Ergebnisse getestet werden, sollte der Status der Testaktivitäten per Kommunikationsplan kommuniziert werden, wie vom Product Owner und Sponsor festgelegt. Ein bekannter Kommunikations Mechanismus ist eine visuelle Abbildung, welche wichtige Informationen in einem einfach zu interpretierenden Format enthält, an einem für alle zugänglichen Ort veröffentlicht wird sowie auf dem neusten Stand ist.

12.1.2.3 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

12.1.3 Outputs

12.1.3.1 Vereinbarung über Funktionierende zu liefernde Ergebnisse*

Abgenommene Arbeitsergebnisse werden vom Kunden und/oder Sponsor formal abgezeichnet und genehmigt. Die formelle Abnahme jedes funktionierenden Arbeitsergebnisses durch den Kunden ist für die Umsatzverbuchung, die Akzeptanz der gesamten Projektergebnisse und die Erfüllung der Projektziele unerlässlich.

12.1.3.2 Funktionierende zu liefernde Ergebnisse*

Dieser Output besteht aus den letztendlichen versandfähigen zu liefernden Ergebnissen, für welche das Projekt sanktioniert wurde. Sobald neue Produkt Inkremente erstellt worden sind, wurden sie kontinuierlich in vorherige Inkremente integriert, sodass ein potenziell versandfertiges Produkt zu jedem Zeitpunkt des Projektes verfügbar ist.

12.1.3.3 Produkt Releases*

Die Produkt Releases sollten Folgendes beinhalten:

- Release-Inhalt—Dies besteht aus wesentlichen Informationen zu den zu liefernden Ergebnissen, die dem Kunden-Support-Team helfen können.
- Freigabemitteilung—Freigabemitteilungen sollten externe oder marktorientierte Lieferkriterien für das zu liefernde Produkt beinhalten.

12.2 Release Rückblick halten

In diesem Prozess, der ein Release abschließt, kommen Business Stakeholdern aus der Organisation und die Mitglieder des Scrum Kernteams zusammen, um auf das Projekt zurückzublicken und Lehren aus dem Projekt zu identifizieren, zu dokumentieren und zu verinnerlichen. Oft führen diese Lehren zur Dokumentation von vereinbarten Maßnahmen, die in zukünftigen Projekten umgesetzt werden.

Abbildung 12-5 zeigt Inputs, Hilfsmittel und Outputs für *Release Rückblick halten* Prozess.

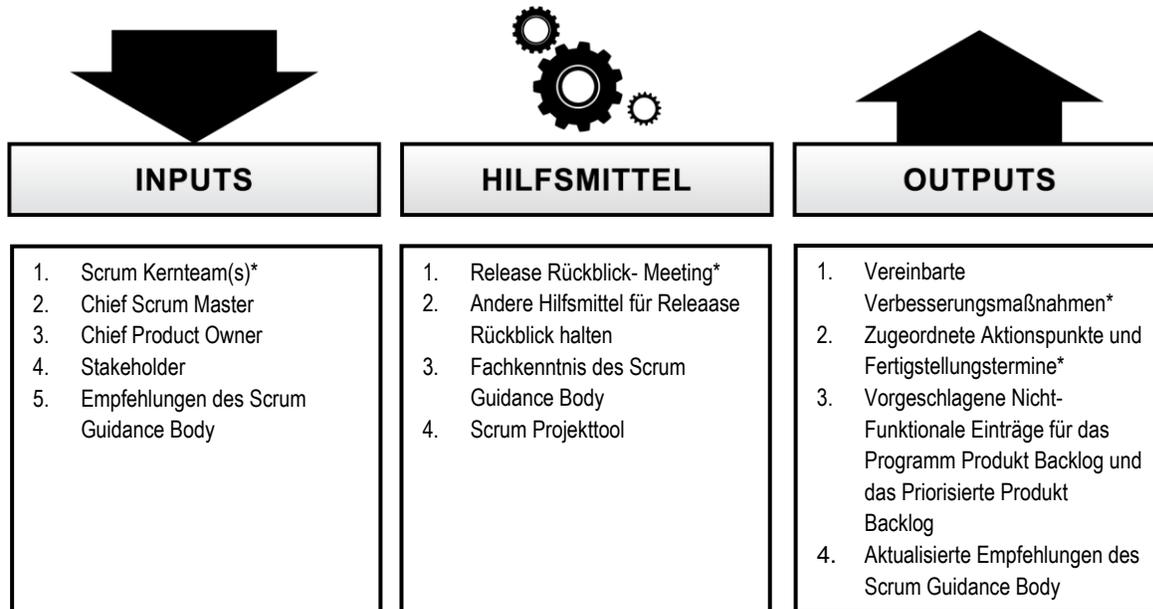


Abbildung 12-5: Release Rückblick halten–Inputs, Hilfsmittel und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

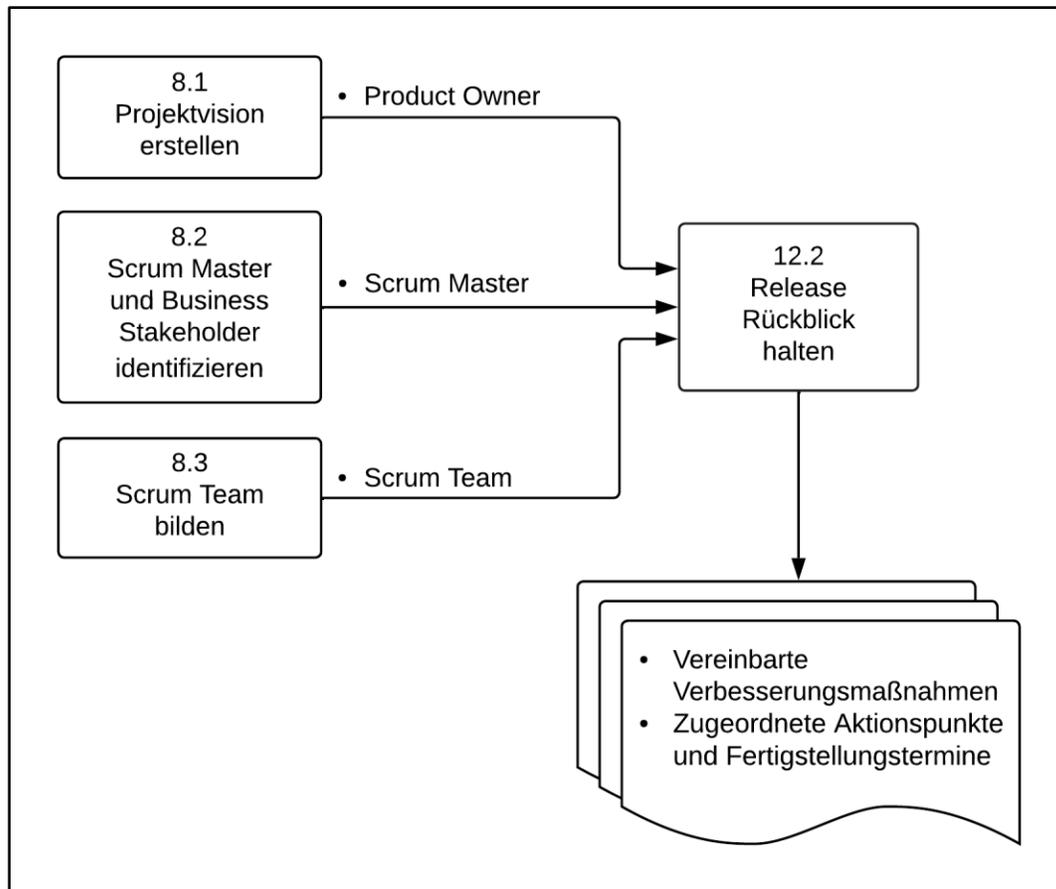


Abbildung 12-6: Release Rückblick halten–Datenflussdiagramm

12.2.1 Inputs

12.2.1.1 Scrum Kernteam(s)*

Siehe Abschnitt 8.4.1.1.

12.2.1.2 Chief Scrum Master

Siehe Abschnitt 3.7.2.2.

12.2.1.3 Chief Product Owner

Siehe Abschnitt 3.7.2.1.

12.2.1.4 Business Stakeholdern

Siehe Abschnitt 8.2.3.2.

12.2.1.5 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11.

Im Prozess *Release Rückblick halten* können die Empfehlungen des Scrum Guidance Body interne Schablonen enthalten, um zukünftige Projekte zu unterstützen und auch einen Leitfaden für das Durchführen des Releaserückblick Meetings. Der zur Verfügung gestellte Leitfaden kann sich auf administratives Prozedere, Audits, Auswertungen und Übergangskriterien beziehen. Häufig beschäftigen sie sich auch damit, wie die Organisation das Wissen, welche auf den gelernten Erfahrungen und Informationen aus allen Projekten stammt, umsetzen will.

12.2.2 Hilfsmittel

12.2.2.1 Release Rückblick- Meeting*

Das Release Rückblick-Meeting ist ein Treffen, um Ansätze festzulegen, wie die Teamzusammenarbeit und die Effektivität für zukünftige Projekte verbessert werden können. Es wird ebenfalls diskutiert, was positiv oder negativ war, und welche potenziellen Verbesserungsmöglichkeiten es gibt. Dieses Treffen ist zeitlich nicht begrenzt (nicht time-boxed) und kann persönlich oder in einem virtuellen Format durchgeführt werden. Zu den Anwesenden gehören das Projekt Team, der Scrum Master, der Chief Scrum Master, der Product Owner, der Chief Product Owner und Business Stakeholdern. Während des Treffens werden die gezogenen Lehren dokumentiert und die Teilnehmer suchen nach Gelegenheiten, um die Prozesse zu verbessern und Ineffizienz zu adressieren.

12.2.2.2 Andere Hilfsmittel, um einen Release Rückblick zu halten

Einige der Hilfsmittel, welche im *Sprint Rückblick halten* Prozess genutzt werden, können auch in diesem Prozess Verwendung finden. Beispiele sind:

- Forscher—Käufer—Urlauber—Gefangener (ESVP) (Siehe Abschnitt 11.2.2.2)
- Schnellboot (Siehe Abschnitt 11.2.2.3)
- Metriken und Messverfahren

12.2.2.3 Fachkenntnis des Scrum Guidance Body

Siehe Abschnitt 8.4.2.7. besprochen.

Im Prozess *Release Rückblick halten* liegt die hauptsächliche Verantwortung des Scrum Guidance Body darin, sicherzustellen, dass die gemachten Erfahrungen eines jeden Projektes nicht verloren gehen und in die Organisation eingebettet werden. Zusätzlich kann dieses Board über Fachkenntnis in verschiedenen Gebieten verfügen, die beim Prozess *Release Rückblick halten* hilfreich sein kann. Dazu gehören beispielsweise Qualität, Personalfragen und Scrum. Es gibt ebenfalls viele Vorschläge in den Empfehlungen des Scrum Guidance Body darüber, wie das Projekt Rückblick-Meeting geführt werden sollte.

12.2.2.4 Scrum Projekttool

Siehe Abschnitt 2.5.3.1.

12.2.3 Outputs

12.2.3.1 Vereinbarte Verbesserungsmaßnahmen*

Siehe Abschnitt 11.3.3.1.

12.2.3.2 Zugeordnete Aktionspunkte und Fertigstellungstermine*

Siehe Abschnitt 11.3.3.2.

12.2.3.3 Vorgeschlagene Nicht-Funktionale Einträge für das Programm Produkt Backlog und das Priorisierte Produkt Backlog

Wenn das ursprüngliche Produkt Backlog oder Priorisierte Produkt Backlog entwickelt werden, basiert es auf User Storys und Anforderungen. Oft sind nicht-funktionalen Anforderungen in den frühen Phasen des Projekts noch nicht vollständig definiert und erscheinen erst während der Sprint Review-, Sprint Rückblick- oder Projekt Rückblick Meetings. Diese nicht-funktionalen Anforderungen sollten dem Programm Produkt Backlog (für das Programm) und dem Priorisierten Produkt Backlog (für das Projekt) hinzugefügt werden sobald sie entdeckt werden. Einige Beispiele von nicht-funktionalen Anforderungen sind Reaktionszeiten, Kapazitätseinschränkungen und sicherheitsbezogene Angelegenheiten.

12.2.3.4 Aktualisierte Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 11.2.3.6.

12.3 Datenflussdiagramm der Release-Phase

In Abbildung 12-7 ist das Datenflussdiagramm für die Release-Phase dargestellt.

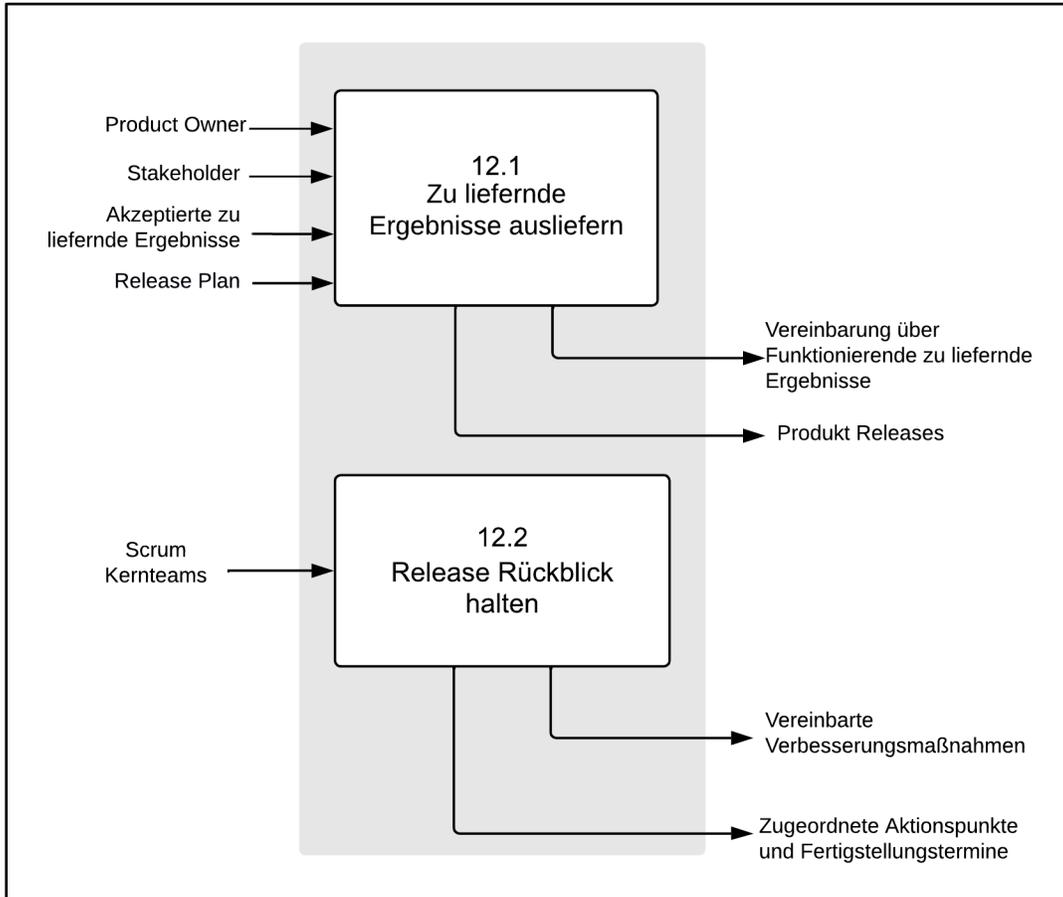


Abbildung 12-7: Releasephase-Datenflussdiagramm

13. Scrum für große Projekte Skalieren

Dieses Kapitel hebt zusätzliche Aspekte von Scrum hervor, die nur für große Projekte zur Anwendung kommen. Die Skalierung von Scrum für große Projekte, wie sie im Scrum Body of Knowledge (SBOK® Guide) definiert ist, ist auf Folgendes anwendbar:

- Große Projekte in jeder Branche
- Produkte, Dienstleistungen oder jegliche an Business Stakeholdern zu liefernde Ergebnisse

Der Begriff "Produkt" im SBOK® Guide kann sich auf ein Produkt, eine Dienstleistung oder andere zu liefernde Ergebnisse beziehen. Scrum kann nicht nur auf kleine Projekte in jeder Branche effektiv angewendet werden, sondern auch auf große, komplexe Projekte mit Hunderten von Team-Mitgliedern und vielen Teams.

Zusätzlich zu den Auswirkungen, die ein großes Projekt auf die fundamentalen Scrum Prozesse in Kapiteln 8-12 haben kann, werden in diesem Kapitel zusätzliche Inputs, Hilfsmittel und Outputs vorgestellt, die auf große Projekte angewendet werden können.

Um die bestmögliche Anwendung des Scrum Rahmens zu ermöglichen, werden in diesem Kapitel zusätzliche Inputs, Hilfsmittel und Outputs entweder als "verbindlich" oder "optional" dargestellt. Die mit Sternchen (*) gekennzeichneten Inputs, Hilfsmittel und Outputs sind verbindlich bzw. werden als kritisch für den Projekterfolg angesehen, während diejenigen ohne Sternchen optional sind.

Es wird empfohlen, dass sich das Scrum Team und diejenigen, die sich erst mit dem Scrum Rahmen und seinen Prozessen vertraut machen, in erster Linie auf die obligatorischen Inputs, Hilfsmitteln und Outputs konzentrieren, während Chief Product Owner, Product Owner, Chief Scrum Master, Scrum Master und Scrum Praktiker mit größerer Erfahrung sich bemühen sollten, sich ein noch tieferes Wissen über die Informationen in diesem gesamten Kapitel anzueignen. Es ist auch wichtig zu wissen, dass, obwohl alle Phasen und Prozesse im SBOK® Guide eindeutig definiert sind, sie nicht unbedingt sequenziell oder separat durchgeführt werden. Je nach den spezifischen Anforderungen des jeweiligen Projekts kann es manchmal sinnvoller sein, dass sich einige Phasen und/oder Prozesse überlappen.

Dieses Kapitel ist aus der Perspektive eines großen Projekts geschrieben, in dem die Aktivitäten mehrerer Scrum Teams koordiniert werden, um gemeinsam potenziell versandfähige Produkt Inkremente / Zu Liefernde Ergebnisse zu erzeugen. Zusätzliche Informationen zur Anwendung von Scrum nicht nur für große Projekte finden Sie in den Kapiteln 2 bis 7, in denen Scrum Prinzipien und Scrum Aspekte behandelt werden.

Großes Scrum Projekt vs. typisches Scrum Projekt

Die in den Kapiteln 8 bis 12 definierten grundlegenden Scrum Prozesse gelten für Scrum Projekte mit einem Product Owner, einem Scrum Master und einem bis drei Scrum Teams. Diese Projekte werden normalerweise als kleine Scrum Projekte angesehen.

Bei großen Projekten, die den Einsatz von vier oder mehr Scrum Teams mit mehreren Product Ownern und mehreren Scrum Mastern erfordern, bleiben die in den Kapiteln 8 bis 12 definierten grundlegenden Prozesse gültig, aber einige zusätzliche Überlegungen und Aktualisierungen der Inputs, Hilfsmitteln und Outputs können erforderlich sein. Dazu gehört zusätzlicher Koordinations- und Synchronisations-Aufwand. Die Auswirkungen auf die grundlegenden Scrum Prozesse bei der Skalierung von Scrum auf große Projekte werden in diesem Kapitel beschrieben.

Die Definition, was ein großes Projekt ist, kann von der Organisation und/oder der Komplexität der durchgeführten Projekte abhängen. Ein wesentliches Kriterium dafür, ob ein Projekt als klein oder groß eingestuft wird, ist, ob das Projekt mehrere Scrum Master und/oder mehrere Product Owner erfordert. Wenn das Projekt nur einen Scrum Master und einen Product Owner benötigt, können diese Personen in der Regel den zusätzlichen Kommunikations- und Synchronisationsaufwand bewältigen, den das Projekt erfordert.

Einige Gründe, warum bei großen Projekten zusätzliche Inputs, Hilfsmittel und Outputs benötigt werden, sind folgende:

Scrum Teams

- Gesteigerte Interaktion und Abhängigkeiten zwischen Scrum Teams, da die Komplexität eines großen Projekts zunimmt
- Die Notwendigkeit, Konflikte zu managen, Probleme zu lösen, Abhängigkeiten zu verwalten und zwischen allen Scrum Teams Prioritäten zu setzen
- Die Notwendigkeit der Spezialisierung, da einige Scrum Teams möglicherweise spezialisierte Ressourcen für bestimmte Aufgaben benötigen (und diese besonderen Fähigkeiten nicht in allen Scrum Teams benötigt werden)
- Die Notwendigkeit, bestimmte Richtlinien und Standards zu definieren, an die sich alle Scrum Teams halten müssen (z.B. Sicherheitsstandards in einem Unternehmen oder gesetzliche und behördliche Richtlinien für bestimmte Branchen); diese können gegebenenfalls vom Scrum Guidance Body definiert werden
- Die Notwendigkeit, eine Umgebung oder einen Arbeitsbereich für das große Projekt einzurichten, die/der dann von allen Scrum Teams genutzt wird
- Die Notwendigkeit, die Outputs mehrerer Scrum Teams zu koordinieren, um die Freigabe eines großen Projekts zu unterstützen

Scrum Master

- Die Notwendigkeit der Zusammenarbeit zwischen Scrum Mastern bei der Beseitigung von Hindernissen und der Synchronisierung der Arbeit der verschiedenen Scrum Teams

Product Owner

- Die Notwendigkeit der Zusammenarbeit zwischen Product Ownern bei der Arbeit mit Business Stakeholdern, der Verfeinerung des priorisierten Produkt Backlogs und der Arbeit mit Scrum Teams.

Es ist auch wichtig zu beachten, dass bei der Skalierung von Scrum für große Projekte zusätzliche unterstützende Dienste wie Architekten, Produktmanager, Compliance, Informationssicherheit, Richtlinien-Gremien usw. benötigt werden können.

13.1 Auswirkungen eines Großen Projekts auf die grundlegenden Scrum Prozesse

Obwohl die grundlegenden Scrum-Prozesse, die in den Kapiteln 8 bis 12 beschrieben wurden, für große Projekte gültig bleiben, gibt es zusätzliche Überlegungen, die beachtet werden sollten. Die Tabellen 13-1 bis 13-5 geben einen Überblick über die Auswirkungen eines großen Projekts auf die grundlegenden Scrum Prozesse für jede Projektphase.

13.1.1 Initiierung

Die Initiierungs-Phase in einem großen Projekt hat die gleichen Ziele und folgt dem gleichen Ablauf wie die Initiierungs-Phase in einem typischen/kleinen Scrum Projekt.

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
8.1 Projektvision erstellen	<p>In diesem Prozess werden ein Chief Product Owner und weitere Product Owner identifiziert. In einem großen Projekt hat der Chief Product Owner die Gesamtverantwortung für das Projekt und arbeitet mit dem Sponsor und anderen Business Stakeholdern zusammen, um die Projektvision zu erstellen. Außerdem wird der Plan für die Zusammenarbeit der Product Owner erstellt.</p> <p>Zusätzlicher Output: Identifizierter Chief Product Owner* (siehe Abschnitt 3.7.2.1)</p> <p>In einem großen Projekt hat der Chief Product Owner die Gesamtverantwortung für das Projekt und arbeitet mit dem Sponsor und anderen Business Stakeholdern zusammen, um die Projektvision zu erstellen. Im Rahmen dieses Prozesses wird der Chief Product Owner identifiziert.</p> <p>Aktualisierter Output: Identifizierte Product Owner* (siehe Abschnitt 8.1.3.1)</p> <p>Da es in einem großen Projekt mehrere Product Owner gibt, müssen diese in diesem Prozess ebenfalls identifiziert werden.</p> <p>Zusätzlicher Output: Product Owner Kollaborationsplan* (siehe Abschnitt 13.2.2)</p> <p>Bei großen Projekten ist es wichtig, dass das gesamte Team der Product Owner Scrum annimmt und für die erfolgreiche Durchführung eines Scrum Projekts zusammenarbeitet.</p>

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
8.2 Scrum Master und Business Stakeholdern identifizieren	<p>In diesem Prozess werden ein Chief Scrum Master und weitere Scrum Master identifiziert. Der Chief Scrum Master konzentriert sich auf die Interaktion und Synchronisation zwischen mehreren Teams. Im Folgenden werden einige zusätzliche oder aktualisierte Ergebnisse für diesen Prozess aufgeführt.</p> <p>Zusätzlicher Output: Identifizierter Chief Scrum Master* (siehe Abschnitt 3.7.2.2)</p> <p>Ähnlich wie der Chief Product Owner sollte auch der Chief Scrum Master für ein großes Projekt identifiziert werden. Er/sie konzentriert sich auf die Interaktion und Synchronisation mehrerer Teams.</p> <p>Aktualisierter Output: Identifizierte Scrum Master*</p> <p>In einem großen Projekt gibt es mehrere Scrum Master, die in diesem Prozess identifiziert werden.</p> <p>Zusätzlicher Output: Organisation für große Scrum Projekte* (siehe Abschnitt 13.2.1)</p> <p>Es ist wichtig, dass das gesamte Projektteam des großen Projekts ein gemeinsames Verständnis von Scrum sowie von Arbeitsvereinbarungen hat, um das Scrum Projekt erfolgreich durchführen zu können.</p> <p>Zusätzlicher Output: Scrum Masters/Scrum Teams Kollaborationsplan* (siehe Abschnitt 13.2.3)</p> <p>Der Scrum Masters/Scrum Teams Kollaborationsplan definiert, wie die Scrum Master und Teams an der Verfeinerung des priorisierten Produkt Backlogs teilnehmen. In diesem Plan wird auch festgelegt, welche Teamvertreter in den Prozess der Verfeinerung einbezogen werden und wie sie ausgewählt werden.</p> <p>Zusätzlicher Output: Geteilte Ressourcen* (siehe Abschnitt 13.2.4)</p> <p>Die Kenntnis der den Scrum Teams zur Verfügung stehenden geteilten Ressourcen ist ein notwendiger Beitrag zur Bildung der einzelnen Scrum Teams.</p> <p>Aktualisierter Output: Identifizierte Unterstützende Dienste (siehe Abschnitt 3.3.2)</p> <p>Zusätzlich zur Identifizierung der normalerweise unterstützenden Dienste können bei einem großen Projekt einige zusätzliche unterstützende Dienste erforderlich sein, um die Aktivitäten zwischen allen Product Ownern, Scrum Mastern und Scrum Teams zu koordinieren.</p>

8.3 Scrum Team bilden

In einem großen Projekt werden mehrere Scrum Teams gebildet, woran mehrere Product Owner und Scrum Master sowie ein Chief Product Owner und ein Chief Scrum Master beteiligt sind. Der Chief Product Owner und der Chief Scrum Master sind an der Bildung der Scrum Teams beteiligt und liefern dabei auch Input bei der Auswahl der Teammitglieder. Der Chief Product Owner und der Chief Scrum Master dienen mehr den Interessen des Gesamtprojekts, während sich die Product Owner und Scrum Master eher auf ihre jeweiligen Scrum Teams konzentrieren.

Zusätzlicher Input: Chief Product Owner* (siehe Abschnitt 3.7.2.1)

Bei einem großen Projekt ist der Chief Product Owner an der Bildung der Scrum Teams beteiligt und hat Einfluss auf die Auswahl der Mitglieder der Teams. Der Chief Product Owner vertritt die Interessen des großen Projekts als Ganzes, während sich die Product Owner auf die Ebene der einzelnen Teams konzentrieren.

Zusätzlicher Input: Chief Scrum Master* (siehe Abschnitt 3.7.2.2)

Bei einem großen Projekt ist der Chief Scrum Master an der Bildung der Scrum Teams beteiligt und hat Einfluss auf die Auswahl der Mitglieder der Teams. Der Chief Scrum Master vertritt die Interessen des großen Projekts als Ganzes, während sich die Scrum Master auf die Ebene der einzelnen Teams konzentrieren.

Zusätzlicher Input: Scrum Masters Kollaborationsplan* (siehe Abschnitt 13.2.3)

Zusätzlicher Input: Team-Spezialisierung* (siehe Abschnitt 13.2.5)

Einige Scrum Teams und Scrum Teammitglieder benötigen möglicherweise spezielle Fähigkeiten, um an bestimmten Problemen im Zusammenhang mit großen Projekten arbeiten zu können.

Zusätzliches Hilfsmittel: Kommunikationsplan für das Große Projekt* (siehe Abschnitt 13.3.1)

Dieser Plan zeigt auf, wie eine effektive Kommunikation zwischen allen an einem Großen Projekt beteiligten Personen zu bewerkstelligen ist.

Zusätzliches Hilfsmittel: Ressourcenplanung für das Große Projekt*

Dieses Hilfsmittel hilft dabei, die Komplexität der Zuordnung verschiedener Arten von Ressourcen an die zahlreichen parallel arbeitenden Scrum Teams zu bewältigen.

Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)

Da an einem großen Projekt viele Personen beteiligt sind und die Komplexität und Interaktion zwischen den Scrum Teams erheblich ist, wird empfohlen, ein Scrum Projekttool zu verwenden, um Prozesse zu automatisieren, die Komplexität zu bewältigen, Berichte zu erstellen, die Kommunikation zwischen den Beteiligten zu handhaben usw.

8.3 Scrum Team bilden (Forts.)	<p>Zusätzliches Hilfsmittel: Identifikation der Umgebung(en)*</p> <p>In großen Projekten ist es wichtig, die Anzahl und die Art der benötigten Umgebungen zu bestimmen, da zahlreiche Scrum Teams gleichzeitig an der Durchführung der Arbeit der jeweiligen Sprints arbeiten werden.</p>
8.4 Epic(s) entwickeln	<p>Jede Interaktion des Product Owners mit dem Kunden und anderen Business Stakeholdern des Unternehmens wird nicht von einem einzelnen Product Owner durchgeführt, sondern vom Chief Product Owner und den mehreren Product Ownern. Wie diese Interaktion aufgeteilt wird, ist im Product Owner Kollaborationsplan definiert. Jegliche Interaktion mit und Beteiligung von Scrum Mastern und/oder Scrum Team Mitgliedern erfolgt so, wie im Scrum Masters Kollaborationsplan definiert. Ansonsten erfolgt die Entwicklung von Epics auf die gleiche Weise wie in einem typischen kleinen Scrum Projekt.</p> <p>Zusätzlicher Input: Product Owner Kollaborationsplan* (siehe Abschnitt 13.2.2)</p> <p>Der Plan definiert, wie die verschiedenen Product Owner untereinander und mit dem Chief Product Owner zusammenarbeiten. Er legt fest, wie sie mit den Business Stakeholdern zusammenarbeiten, um Anforderungen zu sammeln, wie sie das priorisierte Produkt Backlog aktualisieren und wie sie mit den verschiedenen Scrum Teams zusammenarbeiten. Für jedes Scrum Team wird es jeweils nur einen Product Owner geben, der direkt mit ihm zusammenarbeitet. Es muss jedoch festgelegt werden, wie die Scrum Teams auf die Product Owner aufgeteilt werden und mit wie vielen Scrum Teams jeder Product Owner zusammenarbeiten wird.</p> <p>Zusätzlicher Input: Scrum Masters/Scrum Teams Kollaborationsplan* (siehe Abschnitt 13.2.3)</p> <p>Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)</p>

8.5 Priorisiertes Produkt Backlog erstellen	<p>Normalerweise tragen die verschiedenen Product Owner zur Erstellung des priorisierten Produkt Backlogs bei, wie im Product Owner Kollaborationsplan vereinbart. Die endgültige Entscheidung über die Priorisierung und die Lösung von Konflikten zwischen den Product Ownern wird vom Chief Product Owner getroffen. Ansonsten wird der Prozess wie in einem typischen kleinen Scrum Projekt gehandhabt.</p> <p>Zusätzlicher Input: Product Owner Kollaborationsplan* (siehe Abschnitt 13.2.2)</p> <p>Da der Product Owner Kollaborationsplan festlegt, wie die Product Owner Aktualisierungen des priorisierten Produkt Backlogs vornehmen, ist er ein wichtiger Input für diesen Prozess.</p> <p>Zusätzliches Hilfsmittel: Zuordnungen des Priorisierten Produkt Backlogs* (siehe Abschnitt 13.3.4)</p> <p>Da der Chief Product Owner und mehrere Product Owner an einem Großen Projekt beteiligt sind, gewährleisten die Zuordnungen des Priorisierten Produkt Backlogs eine effektive Zuordnung von Epics und User Storys an alle Product Owner.</p> <p>Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)</p>
8.6 Release Planung durchführen	<p>Bei einem Großen Projekt wird der Release Plan vom Chief Product Owner erstellt. Da Releases bei großen Projekten komplizierter sein können, wird im Prozess ein Release Vorbereitungsplan erstellt (falls erforderlich), der später verwendet wird, um für das jeweilige Release zu bestätigen, dass die Anforderungen für das Release erfüllt wurden. Ansonsten wird der Prozess genauso gehandhabt wie in einem typischen Scrum Projekt</p> <p>Zusätzlicher Input: Chief Scrum Master* (siehe Abschnitt 3.7.2.2)</p> <p>Zusätzliches optionales Hilfsmittel: Release Vorbereitungs-Sprint (siehe Abschnitt 13.3.7) Manchmal muss ein Release Vorbereitungs-Sprint geplant werden. Die Anforderungen eines solchen Sprints werden dann in das priorisierte Produkt Backlog aufgenommen.</p> <p>Zusätzlicher Output: Release Vorbereitungsplan* (siehe Abschnitt 13.2.7)</p> <p>Der Release Vorbereitungsplan enthält spezifische Aktivitäten, die kurz vor der Auslieferung bzw. Freigabe durchgeführt werden müssen.</p>

Tabelle 13-1: Auswirkungen eines Großen Projekts auf die grundlegenden Scrum Prozesse—Initiierungs-Phase

13.1.2 Planen und Abschätzen

Wie in einem typischen/kleinen Scrum Projekt hat die Planungs- und Abschätzungsphase in einem Großen Projekt das Ziel, sich auf die im kommenden Sprint zu erledigende Arbeit zu einigen und diese zu planen. Jedes Scrum Team arbeitet mit seinem jeweiligen Scrum Master und Product Owner zusammen, um die spezifische Arbeit für den Sprint festzulegen und zu planen, wie das Team die Arbeit durchführen wird (wie in einem typischen Scrum Projekt).

Im Vergleich zu einem kleinen Scrum Projekt müssen zusätzliche Schritte unternommen werden, um die Arbeit zwischen den verschiedenen Scrum Teams aufzuteilen. Welche Teams und Teammitglieder an der Erstellung und Abschätzung welcher User Story beteiligt sind und wie die Zusage von User Storys und die Erstellung der jeweiligen zu Liefernden Ergebnisse zwischen den verschiedenen Teams aufgeteilt werden, basiert auf der Spezialisierung der Teams und dem Kollaborationsplan.

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
9.1 User Storys erstellen	<p>In einem großen Projekt sind mehrere Product Owner und mehrere Scrum Teams an der Erstellung der User Storys beteiligt. Nicht jeder Product Owner und nicht jedes Scrum Team/Scrum Team Mitglied kann an der Erstellung jeder User Story beteiligt sein. Daher wird die Erstellung von User Storys zwischen mehreren Product Ownern auf der Grundlage des Product Owner Kollaborationsplans sowie zwischen mehreren Scrum Teams auf der Grundlage der Team-Spezialisierung und des Scrum Masters/Scrum Teams Kollaborationsplans aufgeteilt. Ansonsten läuft der Prozess so ab wie in einem typischen kleinen Scrum Projekt.</p> <p>Zusätzlicher Input: Product Owner Kollaborationsplan* (siehe Abschnitt 13.2.2)</p> <p>Zusätzlicher Input: Scrum Masters/Scrum Teams Kollaborationsplan* (siehe Abschnitt 13.2.3)</p> <p>Zusätzliches Hilfsmittel: Identifizierung der Umgebung(en)* (siehe Abschnitt 13.3.3)</p> <p>In einem großen Projekt ist es wichtig, die Anzahl und die Art der Umgebungen zu ermitteln, die für eine effektive Arbeit des Scrum Teams erforderlich sind.</p> <p>Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)</p>

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
9.2 User Storys abschätzen	<p>In einem großen Projekt kann nicht jedes Scrum Team/jedes Mitglied eines Scrum Teams an der Abschätzung jeder User Story beteiligt sein. Daher beteiligen sich in der Regel nur bestimmte Teams/Teammitglieder an der Abschätzung von User Storys. Welches Team bzw. welche Teammitglieder welche User Storys abschätzen, wird anhand der Team-Spezialisierung und des Scrum Masters/Scrum Teams Kollaborationsplans entschieden.</p> <p>Welcher Product Owner mit den jeweiligen Scrum Teams/Teammitgliedern zusammenarbeitet, hängt in der Regel davon ab, wer die jeweiligen User Storys erstellt hat und am besten in der Lage ist, alle Fragen zu den abzuschätzenden User Storys zu beantworten. Das basiert ebenfalls auf dem Product Owner Kollaborationsplan.</p> <p>Ansonsten erfolgt die Schätzung der User Storys auf die gleiche Weise wie in einem typischen Scrum Projekt.</p> <p>Zusätzlicher Input: Product Owner Kollaborationsplan* (siehe Abschnitt 13.2.2)</p> <p>Zusätzlicher Input: Scrum Masters/Scrum Teams Kollaborationsplan* (siehe Abschnitt 13.2.3)</p> <p>Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)</p>
9.3 User Stories zusagen	<p>Für jeden Sprint in einem großen Projekt wird jedes Team gebeten, eine bestimmte Teilmenge von User Storys für den Sprint zuzusagen und dann die entsprechenden zu Liefernden Ergebnisse zu erstellen. Die Entscheidung, welches Team welche User Storys umsetzen soll, hängt von den spezifischen Fähigkeiten des jeweiligen Teams ab und basiert auf der Spezialisierung des Teams.</p> <p>Ansonsten wird die Zusage der User Storys genauso gehandhabt wie in einem typischen kleinen Scrum Projekt, basierend auf Prioritäten, Abschätzungen und teamspezifischer Geschwindigkeit.</p> <p>Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)</p>

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
9.4 Aufgaben Identifizieren	<p>Die Identifizierung von Aufgaben wird von jedem Team durchgeführt, um die User Storys, zu denen sich das Team verpflichtet hat, weiter zu zerlegen. Jedes Scrum Team identifiziert die Aufgaben auf ähnliche Weise wie in einem typischen kleinen Scrum Projekt. Die korrekte Identifizierung von Abhängigkeiten hilft den Scrum Teams jedoch dabei, festzustellen, welche ihrer Entscheidungen und Aktivitäten sich auf andere Teams auswirken können, und kann auch die relative Reihenfolge beeinflussen, in der ein Scrum Team seine jeweiligen Aufgaben ausführt, um die zu Liefernden Ergebnisse des Sprints zu erstellen.</p> <p>Aktualisiertes Hilfsmittel: Abhängigkeitsermittlung*</p> <p>Beschrieben in Abschnitt 9.4.2.3. In großen Projekten hilft die korrekte Identifizierung von Abhängigkeiten den Scrum Teams herauszufinden, welche ihrer Entscheidungen und Aktivitäten sich auf andere Teams auswirken können. Dadurch kann auch die relative Reihenfolge beeinflusst werden, in der ein einzelnes Scrum Team seine jeweiligen Aufgaben ausführt, um die zu Liefernden Ergebnisse des Sprints zu erstellen.</p> <p>Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)</p>
9.5 Aufgaben abschätzen	<p>Die im vorangegangenen Prozess identifizierten Aufgaben können optional abgeschätzt werden, genau wie es in einem typischen kleinen Scrum Projekt gemacht wird.</p> <p>Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)</p>
9.6 Sprint Backlog aktualisieren	<p>Jedes Team aktualisiert sein spezifisches Sprint Backlog entsprechend seiner zugesagten User Storys und zugehörigen Aufgaben. Jedes Team aktualisiert das Sprint Backlog genauso, wie es in einem typischen kleinen Scrum Projekt gemacht wird.</p> <p>Zusätzlicher Input: Umgebung und Umgebungsplan (siehe Abschnitt 13.2.6)</p> <p>Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)</p>

Tabelle 13-2: Auswirkungen eines Großen Projekts auf die grundlegenden Scrum Prozesse—Planungs- und Abschätzungs-Phase

13.1.3 Implementieren

In der Implementierungsphase eines großen Projekts erstellt jedes Scrum Team, unterstützt von seinem Scrum Master, die zu Liefernden Ergebnisse, die mit den User Storys verbunden sind, zu denen es sich verpflichtet hat. Dazu werden die Aufgaben bearbeitet und abgeschlossen, die das Team in der Planungs- und Abschätzungsphase identifiziert hat.

Im Vergleich zu einem kleinen Scrum Projekt müssen zusätzliche Schritte unternommen werden, um sicherzustellen, dass eine effektive Kommunikation zwischen den verschiedenen Teams stattfindet (wie im Kommunikationsplan definiert) und dass die Arbeit synchronisiert wird (wie im Kollaborationsplan beschrieben).

Während die Scrum Teams die zu Liefernden Ergebnisse des Sprints erstellen, verfeinern der Chief Product Owner und die anderen Product Owner das priorisierte Produkt Backlog, um es bei Änderungen der Anforderungen und/oder Änderungen der Prioritäten auf dem neuesten Stand zu halten. Sie stellen auch sicher, dass die User Storys, die die Scrum Teams im nächsten Sprint liefern sollen, für die Abschätzung und die Zusage bereit sind. Das erfolgt so wie in einem typischen/kleinen Scrum Projekt.

Bei einem großen Projekt teilen die Product Owner ihre Arbeit auf der Grundlage des Product Owners Kollaborationsplan auf. Die Interaktion mit den Scrum Teams und/oder Scrum Team Mitgliedern basiert auf der Spezialisierung der Scrum Teams und dem Kollaborationsplan.

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
10.1 Zu Liefernde Ergebnisse erzeugen	<p>Jedes Team erzeugt die zu Liefernden Ergebnisse, die mit den User Storys verbunden sind, zu denen es sich verpflichtet hat.</p> <p>Anders als bei einem typischen kleinen Scrum Projekt ist ein Scrum Team in einem großen Projekt in seiner Freiheit, seine Arbeit zu organisieren, eingeschränkt. Diese Einschränkungen ergeben sich aus der Notwendigkeit, Ressourcen mit anderen Teams zu teilen (wie in den geteilten Ressourcen angegeben und vereinbart), sowie aus der Umgebung und dem Umgebungsplan und auch aus den potenziellen Auswirkungen von Entscheidungen, die andere Teams betreffen könnten.</p> <p>Darüber hinaus sind zusätzliche Aktivitäten erforderlich, um eine gute Kommunikation und Synchronisation zwischen den Scrum Teams zu gewährleisten.</p> <p>Zusätzlicher Input: Release Vorbereitungsplan* (siehe Abschnitt 13.2.7)</p> <p>Zusätzliches Hilfsmittel: Scrum der Scrums Meetings* (siehe Abschnitt 13.3.5)</p> <p>Hierbei handelt es sich um fokussierte Meetings, bei denen Vertreter der Scrum Teams zusammenkommen, um den Status ihrer jeweiligen Teams auszutauschen.</p> <p>Zusätzliches Hilfsmittel: Release-Vorbereitungsmethoden* (siehe Abschnitt 13.3.6)</p> <p>Release-Vorbereitungsmethoden werden verwendet, um die im Release Vorbereitungsplan für die Vorbereitung der zu liefernden Produkte für die Auslieferung/Freigabe identifizierten Aufgaben auszuführen.</p>
10.2 Tägliches Standup durchführen	<p>Jedes Team führt sein Tägliches Standup Meeting genauso durch, wie es in einem typischen Scrum Projekt durchgeführt wird. Da jedoch jeder Scrum Master möglicherweise mit mehreren Scrum Teams arbeitet, ist ein gewisser Koordinationsaufwand erforderlich, um Terminkonflikte zwischen den Täglichen Standup Meetings der jeweiligen Scrum Teams zu vermeiden.</p> <p>Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)</p>

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
10.3 Priorisiertes Produkt Backlog verfeinern	<p>Die Unterschiede im Vergleich zu einem typischen kleinen Scrum Projekt sind die gleichen wie in den Prozessen Epics entwickeln und Priorisiertes Produktbacklog erstellen. Insbesondere wird jede Interaktion des Product Owners mit dem Kunden und anderen Business Stakeholdern vom Chief Product Owner und/oder den mehreren Product Ownern und nicht von einem einzelnen Product Owner durchgeführt. Wie diese Interaktion aufgeteilt wird, ist im Product Owner Kollaborationsplan festgelegt. Außerdem werden die endgültigen Entscheidungen über die Priorisierung vom Chief Product Owner getroffen.</p> <p>Jegliche Interaktion mit und Teilnahme von Scrum Mastern und/oder Scrum Team Mitgliedern erfolgt so, wie im Scrum Masters/Scrum Teams Kollaborationsplan definiert. Ansonsten wird die Verfeinerung des priorisierten Produkt Backlogs wie in einem typischen Scrum Projekt gehandhabt.</p> <p>Zusätzlicher Input: Product Owner Kollaborationsplan* (siehe Abschnitt 13.2.2)</p> <p>Der Product Owner Kollaborationsplan legt fest, wie die Product Owner Aktualisierungen am Priorisierten Produkt Backlog vornehmen.</p> <p>Zusätzlicher Input: Scrum Masters Kollaborationsplan* (siehe Abschnitt 13.2.3)</p> <p>Zusätzlicher Input: Team-Spezialisierung* (siehe Abschnitt 13.2.5)</p> <p>Zusätzliches optionales Hilfsmittel: Scrum Projekttool (siehe Abschnitt 13.3.8)</p> <p>Zusätzliches optionales Hilfsmittel: Release Vorbereitungs-Sprint* (siehe Abschnitt 13.3.7)</p> <p>Zusätzliche Output: Aktualisierter Release Vorbereitungsplan* (siehe Abschnitt 13.2.7)</p> <p>Änderungen am priorisierten Produkt Backlog, die während der Verfeinerung des priorisierten Produkt Backlogs vorgenommen werden, können sich auf den Release Vorbereitungsplan auswirken.</p>

Tabelle 13-3: Auswirkungen eines Großen Projekts auf die grundlegenden Scrum Prozesse—Implementierungs-Phase

13.1.4 Review und Rückblick

In der Review- und Rückblicks-Phase eines Großen Projekts führt jedes Scrum Team seinem jeweiligen Product Owner die vom Team in einem Sprint erstellten Ergebnisse zur Genehmigung und zum Feedback vor und trifft sich dann, um Wege zur kontinuierlichen Verbesserung seiner Arbeit zu finden. Dies geschieht auf die gleiche Weise wie in einem typischen Scrum Projekt.

Zusätzlich treffen sich die Product Owner, um Wege zur Verbesserung ihrer Arbeit zu finden. Als letzter Schritt eines Sprints treffen sich der Chief Product Owner und der Chief Scrum Master mit den entsprechenden Product Ownern, Scrum Mastern und Scrum Team Mitgliedern, um die Ergebnisse ihrer jeweiligen Sprint Retrospektions-Meetings auszutauschen. Da jeder Scrum Master und jeder Product Owner mit mehreren Scrum Teams zusammenarbeiten kann, ist ein gewisser Koordinationsaufwand erforderlich, um Terminkonflikte zwischen den Sprint Review Meetings und/oder den Sprint Retrospektions-Meetings der verschiedenen Scrum Teams zu vermeiden.

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
11.1 Sprint vorführen und validieren	<p>Dieser Prozess wird von jedem Scrum Team individuell durchgeführt. Für jedes Team genehmigt der jeweilige Product Owner die fertigen User Storys. Dies kann jedoch aufgrund von gegenseitigen Abhängigkeiten etwas komplex sein. Es kann vorkommen, dass der Chief Product Owner an Sprint Review Meetings für einige Scrum Teams teilnimmt, die voneinander abhängige Zu Liefernde Ergebnisse haben.</p> <p>Zusätzlicher Input: Chief Product Owner* (siehe Abschnitt 3.7.2.1)</p>

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
11.2 Sprint Rückblick halten	<p>Jedes Scrum Team im Projekt trifft sich mit seinem Scrum Master zu einem Sprint Retrospektions-Meeting, das in der gleichen Weise wie in einem typischen kleinen Scrum Projekt durchgeführt wird.</p> <p>Da ein einzelner Scrum Master und ein einzelner Product Owner mit mehreren Scrum Teams arbeiten können, ist ein gewisser Koordinationsaufwand erforderlich, um Terminkonflikte zwischen den Sprint Retrospektions-Meetings verschiedener Scrum Teams zu vermeiden.</p> <p>Außerdem treffen sich der Chief Product Owner und die Product Owner zu einem Sprint Retrospektions-Meeting, um ihre Zusammenarbeit und andere Aspekte des Sprints zu besprechen.</p> <p>Zusätzlich treffen sich Scrum Master und/oder andere Vertreter aus jedem Scrum Team zu einem speziellen Scrum der Scrums Meeting, um Best Practices und andere Ergebnisse der Sprint Retrospektions-Meetings der verschiedenen Teams auszutauschen, z.B. Probleme bei der Zusammenarbeit zwischen den Teams. Da in einem großen Projekt sehr oft Best Practices und Probleme aus der Zusammenarbeit zwischen den verschiedenen Scrum Teams und dem Team der Product Owner entstehen, ist es üblich, dass der Chief Product Owner und andere Product Owner an diesem Meeting teilnehmen.</p> <p>Zusätzlicher Input: Product Owner Kollaborationsplan* (siehe Abschnitt 13.2.2)</p> <p>Die Verfeinerung des priorisierten Produkt Backlogs kann sich bei großen Projekten als besonders schwierig erweisen. Wenn sie nicht effektiv durchgeführt wird, kann sie zu Problemen führen und den Aufwand für die Teams erhöhen. Daher wird empfohlen, in den Retrospektions-Meetings auch über die Verfeinerung zu sprechen und sich dabei besonders darauf zu konzentrieren, wie die verschiedenen Product Owner miteinander und mit den Scrum Teams zusammenarbeiten, um so eine hohe Effektivität der Verfeinerung des Backlogs zu erreichen. Außerdem sollten Epics und User Storys mit vielen Abhängigkeiten in Gruppen zusammengefasst werden.</p> <p>Zusätzlicher Input: Scrum Masters Kollaborationsplan* (siehe Abschnitt 13.2.3.)</p> <p>Die Verfeinerung des priorisierten Produkt Backlogs kann sich bei großen Projekten als besonders schwierig erweisen. Wenn sie nicht effektiv durchgeführt wird, kann sie zu Problemen führen und den Aufwand für die Teams erhöhen. Daher wird empfohlen, in den Retrospektions-Meetings auch über die Verfeinerung zu sprechen, mit besonderem Augenmerk auf das Zusammenspiel zwischen den verschiedenen Scrum Mastern, Scrum Teams und deren Interaktion mit den Product Ownern bei den Verfeinerungsaktivitäten.</p>

Tabelle 13-4: Auswirkungen von großen Projekten auf die grundlegenden Scrum-Prozesse - Review- und Rückblicks-Phase

13.1.5 Release

In der Release-Phase eines großen Projekts stimmen sich die Product Owner der einzelnen Scrum Teams mit dem Chief Product Owner ab, um sicherzustellen, dass die Ergebnisse aller Scrum Teams synchronisiert, integriert und wie vom Kunden und anderen Business Stakeholdern gefordert freigegeben werden.

Darüber hinaus werden der Product Owner Kollaborationsplan und der Scrum Masters/Scrum Teams Kollaborationsplan verfeinert, um eine bessere Koordination zwischen allen Teams in einem großen Projekt zu gewährleisten.

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
12.1 Zu Liefernde Ergebnisse ausliefern	<p>In einem großen Projekt werden alle akzeptierten Zu Liefernden Ergebnisse aus den zuvor abgeschlossenen Sprints in der Regel auf dieselbe Weise wie in einem typischen kleinen Scrum Projekt an die Business Stakeholdern geliefert oder übergeben. Aufgrund von geschäftlichen Erwägungen und der Komplexität großer Projekte können jedoch zusätzliche Vorbereitungsschritte für die Freigabe erforderlich sein.</p> <p>Zusätzlicher Input: Chief Product Owner (siehe Abschnitt 3.7.2.1)</p> <p>Zusätzlicher Input: Leitender Scrum Master (siehe Abschnitt 3.7.2.2)</p> <p>Zusätzlicher Input: Release-Vorbereitungsplan (siehe Abschnitt 13.2.7)</p>

Prozess	Zusammenfassung der Auswirkungen eines Großen Projekts
12.2 Release Rückblick halten	<p>Zusätzlicher Input: Chief Product Owner* (siehe Abschnitt 3.7.2.1)</p> <p>Zusätzlicher Input: Chief Scrum Master* (siehe Abschnitt 3.7.2.2)</p> <p>Zusätzlicher Input: Product Owner Kollaborationsplan* (siehe Abschnitt 13.2.2)</p> <p>Die Verfeinerung des priorisierten Produkt Backlogs kann sich bei großen Projekten als besonders schwierig erweisen. Wenn sie nicht effektiv durchgeführt wird, kann sie zu Problemen führen und den Aufwand für die Teams erhöhen. Daher wird empfohlen, in den Retrospektions-Meetings auch über die Verfeinerung zu sprechen und sich dabei besonders darauf zu konzentrieren, wie die verschiedenen Product Owner miteinander und mit den Scrum Teams zusammenarbeiten, um so eine hohe Effektivität der Verfeinerung des Backlogs zu erreichen. Außerdem sollten Epics und User Storys mit vielen Abhängigkeiten in Gruppen zusammengefasst werden.</p> <p>Zusätzlicher Input: Scrum Masters/Scrum Teams Kollaborationsplan* (siehe Abschnitt 13.2.3)</p> <p>Die Verfeinerung des priorisierten Produkt Backlogs kann sich bei großen Projekten als besonders schwierig erweisen. Wenn sie nicht effektiv durchgeführt wird, kann sie zu Problemen führen und den Aufwand für die Teams erhöhen. Daher wird empfohlen, in den Retrospektions-Meetings auch über die Verfeinerung zu sprechen, mit besonderem Augenmerk auf das Zusammenspiel zwischen den verschiedenen Scrum Mastern, Scrum Teams und deren Interaktion mit den Product Ownern bei den Verfeinerungsaktivitäten.</p>

Tabelle 13-5: Auswirkungen von großen Projekten auf die grundlegenden Scrum-Prozesse - Release-Phase

13.2 Zusätzliche Inputs und Outputs für große Projekte

13.2.1 Organisation für große Scrum Projekte*

Die geeignete Organisationsstruktur, die für die Durchführung und Unterstützung großer Projekte erforderlich ist, wird in Absprache mit dem Chief Product Owner, dem Chief Scrum Master, Leitenden Angestellten, dem Scrum Guidance Body und anderen relevanten Experten festgelegt.

Organisationen, die planen, Scrum für die Umsetzung großer Projekte zu verwenden, sollten das Scrum-Framework vollständig übernehmen. Die Organisation sollte in der Lage sein, die erforderlichen Ressourcen zur Verfügung zu stellen. Wenn die Organisation nicht in der Lage ist, die erforderlichen Ressourcen bereitzustellen, müssen Pläne für die Beschaffung der notwendigen Ressourcen wie Mitarbeiter, Hilfsmittel und Arbeitsräume gemacht werden. Eine Organisation, die den Einsatz von Scrum plant, muss unbedingt darauf vorbereitet sein, ihre Arbeitskultur und -gewohnheiten drastisch zu ändern, um die Vorteile von Scrum wirklich nutzen zu können.

In einer Scrum-Umgebung, in der große Projekte entwickelt werden, gibt es zahlreiche Scrum Teams, die um Ressourcen konkurrieren. Daher ist es wichtig, die organisatorischen Ressourcen optimal zu verwalten, um die übergreifenden Projektziele zu erreichen. Die Scrum-Organisation für ein großes Projekt sollte u. A. Folgendes berücksichtigen: die Komponenten, die entwickelt werden sollen; die Fähigkeiten, Kosten und anderen Ressourcen, die für ihre Entwicklung erforderlich sind; die aktuellen Geschwindigkeiten der Scrum Teams (oder geschätzte/angenommene Geschwindigkeiten), um eine grobe Schätzung der Projektdauer zu erhalten; die Kommunikationsanforderungen und andere Schnittstellen, die die Scrum Teams pflegen müssen.

13.2.2 Product Owner Kollaborationsplan*

Der Chief Product Owner arbeitet mit den wichtigsten Product Ownern zusammen, um den Product Owners Kollaborationsplan zu erstellen. Der Product Owner Kollaborationsplan sollte genau definieren, wie die Product Owner mit dem Chief Product Owner zusammenarbeiten sollen. In diesem Plan sollte zumindest festgelegt werden, mit wie vielen Scrum Teams ein Product Owner zusammenarbeiten kann (in der Regel auf der Grundlage von Faktoren wie Erfahrung, Zeit und Domänenwissen), wie die Aufgabe der Erfassung von Anforderungen der Business Stakeholdern unter den Product Ownern aufgeteilt wird, wie das priorisierte Produkt Backlog mit neuen Anforderungen oder Änderungen der Anforderungen aktualisiert wird und wie die Product Owner mit mehreren Scrum Teams zusammenarbeiten werden. Es ist zu beachten, dass jedes Scrum Team weiterhin nur mit einem Product Owner zusammenarbeitet; ein einzelner Product Owner kann jedoch bei Bedarf mit mehr als einem Scrum Team zusammenarbeiten.

13.2.3 Scrum Masters/Scrum Teams Kollaborationsplan*

Der Chief Scrum Master arbeitet mit den wichtigsten Scrum Mastern, dem Scrum Guidance Body und manchmal auch mit einigen bereits identifizierten Mitgliedern der Scrum Teams zusammen, um den Scrum Masters Kollaborationsplan zu erstellen. Der Scrum Masters Kollaborationsplan definiert, wie die zahlreichen Scrum Teams zusammenarbeiten werden, um den höchsten Mehrwert in der kürzest möglichen Zeit zu liefern.

In einem großen Projekt werden in der Regel mehrere Scrum Master eingesetzt, die jeweils ein produktives Arbeitsumfeld für ihr jeweiliges Scrum Team schaffen und sicherstellen. Es ist möglich, dass ein Scrum Master mehr als ein Scrum Team zur gleichen Zeit unterstützt. Scrum Master müssen mit anderen Scrum Mastern sowie dem Chief Scrum Master, dem Chief Product Owner und den Product Ownern zusammenarbeiten, um die Liste der Komponenten und Ressourcen zu erstellen, die für alle Teams während des gesamten Projekts gemeinsam benötigt werden. Sie tragen auch dazu bei, Inputs für die Erstellung des Release Vorbereitungsplans zu liefern.

Der Plan sollte Informationen über spezialisierte Bereiche enthalten, die qualifizierten Teams zugeordnet werden, wie die Teams die Verfeinerung des priorisierten Produkt Backlogs und die Abschätzung unterstützen werden (d. h. welche Teammitglieder an den Verfeinerungssitzungen und Abschätzungen teilnehmen werden) und wie die Teams die Scrum der Scrums Meetings organisieren werden. Es kann auch notwendig sein, ein Scrum Projekttool zu verwenden, um die Anwendung von Scrum in großen Projekten zu erleichtern.

Der Scrum Masters/Scrum Teams Kollaborationsplan kann auch Informationen darüber enthalten, wie die einzelnen Scrum Teams gecoacht werden sollen. Er kann zum Beispiel Informationen darüber enthalten, ob es neben dem Scrum Master einen separaten Coach geben wird; ob es bei verteilten Teams an jedem Standort einen Scrum Master geben wird; wie die Teammitglieder mit Scrum Mastern, die sich an einem anderen Ort befinden, und mit Scrum Mastern, die sich nicht an einem anderen Ort befinden, zusammenarbeiten werden, usw.

Obwohl Scrum Teams, die an einem großen Projekt arbeiten, bei der Erstellung des priorisierten Produkt Backlogs miteinander interagieren müssen, können Product Owner sicherstellen, dass User Storys mit vielen Abhängigkeiten gruppiert werden und einem einzelnen Product Owner gehören, so dass die Ergebnisse nur von einem oder wenigen Scrum Teams bearbeitet werden. Dies minimiert die Abhängigkeiten von Aufgaben zwischen verschiedenen Scrum Teams, so dass jedes Scrum Team effizienter arbeiten kann, um seine Ergebnisse zu erstellen.

13.2.4 Geteilte Ressourcen*

Zu den geteilten Ressourcen können Personal, Umgebungen und Ausrüstung gehören, die von einigen oder allen am Projekt arbeitenden Scrum Teams benötigt werden. In einem großen Projekt sind die geteilten Ressourcen begrenzt und werden möglicherweise von mehreren Scrum Teams zur gleichen Zeit benötigt. Vor diesem Hintergrund müssen sich der Chief Product Owner, der Chief Scrum Master, andere Product Owner und andere Scrum Master auf eine Methode einigen, wie geteilte Ressourcen zugewiesen werden. Eine mögliche Vorgehensweise zur Zuweisung geteilter Ressourcen könnte sein, sicherzustellen, dass die Ressourcen zuerst den Teams zugeteilt werden, die an den wichtigsten/wertvollsten Features arbeiten.

Wenn konkurrierende Anforderungen sehr ähnliche Prioritäten haben, sollte der Chief Product Owner die Ressourcenzuteilung auf der Grundlage der aktuellen Geschäftlichen Anforderungen, Prioritäten und anderer definierter Kriterien entscheiden.

13.2.5 Team-Spezialisierung*

Es gibt drei Dimensionen der Team-Spezialisierung, die bei der Anwendung von Scrum auf große Projekte berücksichtigt werden sollten. Die erste Dimension ist die Notwendigkeit, eine Reihe spezifischer Aufgaben zu erledigen. Ein spezialisiertes Team könnte z. B. ein Integrationsteam sein, das über Expertenwissen im Bereich der kontinuierlichen Integration verfügt. Dieses Wissen könnte besonders wichtig sein, wenn ein Release Vorbereitungs-Sprint durchgeführt wird (wenn bestimmte Aufgaben vor einer Freigabe durchgeführt werden müssen).

Die zweite Dimension ist der Bedarf an speziellen Fertigkeiten von einzelnen Teammitgliedern. Theoretisch sind alle Mitglieder des Scrum Teams Generalisten und Spezialisten, d. h. sie verfügen über allgemeine Kenntnisse in verschiedenen Bereichen und sind Experten in mindestens einem. Bei einem großen Projekt ist dies jedoch möglicherweise nicht der Fall. Die Teammitglieder müssen unter Umständen über sehr spezifische Fertigkeiten verfügen - z. B. ein spezielles Fachwissen wie Sicherheit -, die möglicherweise nicht in allen Teams des großen Projekts verfügbar sind, für die sie erforderlich wären. In diesem Fall wäre es äußerst kostspielig, alle Mitarbeiter in allen erforderlichen Spezialgebieten zu schulen.

Experten mit speziellen Fertigkeiten und Kenntnissen können dem Scrum Projekt zugewiesen werden und je nach Bedarf in verschiedenen Teams eingesetzt werden. Manchmal kann es jedoch notwendig sein, bei Bedarf Experten aus externen Quellen zu akquirieren, wobei zu bedenken ist, dass das Hinzufügen eines neuen Teammitglieds die Geschwindigkeit des Teams beeinträchtigen wird.

Die dritte Dimension für die Akquirierung von Experten ergibt sich aus den Einschränkungen hinsichtlich der Flexibilität von Teams. Typischerweise hat jedes Team in einem Scrum Projekt einen oder mehrere Bereiche, in denen es über ein beträchtliches Fachwissen verfügt, ein paar Bereiche, die es mit etwas Input und Training bearbeiten kann, und andere Bereiche, in denen es nicht über die nötigen Fähigkeiten oder Erfahrungen verfügt, um sinnvoll daran arbeiten zu können. Während der Sprint-Planung wird es für jedes Team eine Teilmenge von User Storys geben, die logischerweise dem Team auf der Grundlage seiner Fachkenntnisse zugewiesen werden, einige, an denen es arbeiten kann, und einige, an denen es nicht arbeiten kann, weil es nicht über die erforderlichen Kenntnisse oder Fähigkeiten verfügt. In diesem Fall müssen möglicherweise Experten hinzugezogen werden, wenn die Fähigkeiten der Teammitglieder nicht flexibel genug sind.

Der begrenzte Zugang zu Ressourcen mit speziellen Fähigkeiten in einem großen Scrum Projekt birgt ein gewisses Risiko für das Projekt. Einige User Storys von sehr hoher Priorität können möglicherweise deshalb in einem spezifischen Sprint nicht fertiggestellt werden. Die Teams müssen möglicherweise an einigen User Storys mit niedrigerer Priorität arbeiten, während sie auf die Verfügbarkeit von Teammitgliedern mit speziellen Fähigkeiten warten.

13.2.6 Umgebung und Umgebungsplan*

Die erforderlichen Entwicklungsumgebungen sind möglicherweise während der gesamten Dauer eines großen Projekts nicht immer verfügbar. Beispielsweise kann ein Kran, der für eine Bautätigkeit benötigt wird, oder eine spezielle Testumgebung nur an bestimmten Tagen verfügbar sein. Sobald alle erforderlichen Umgebungen identifiziert sind, wird ein Umgebungsplan erstellt und für die Koordinierung der Sprint-Aktivitäten im Prozess "Sprint Backlog aktualisieren" verwendet. Der Umgebungsplan ist ein Kalender, der detailliert festlegt, wie die Scrum Teams auf bestimmte Umgebungen zugreifen und diese gemeinsam nutzen werden. Der Zeitplan weist Tage und Zeiträume zu, in denen jedes Team jede Umgebung nutzen kann.

13.2.7 Release Vorbereitungsplan*

Der Chief Product Owner arbeitet mit anderen Product Ownern zusammen, um den Release Vorbereitungsplan zu erstellen. Der Release Vorbereitungsplan beschreibt die Schritte, die von den relevanten Scrum Teams und anderen Personen unternommen werden müssen, um zu bestätigen, dass die Mindestanforderungen für die Freigabe erfüllt sind und das Produkt oder Produktinkrement bereit für die Freigabe ist. Die geschäftlichen Entscheidungen mit den dazugehörigen geschäftlichen Begründungen für die Durchführung von Release-Vorbereitungsaufgaben werden ebenfalls im Release-Vorbereitungsplan beschrieben.

Da jeder Sprint ein potenziell versandfähiges Produkt oder ein anderes zu Lieferndes Ergebnis erzeugt, kann in einem typischen kleinen Scrum Projekt eine Freigabe nach jedem Sprint erfolgen, wenn es geschäftlich sinnvoll ist. In einem großen Scrum Projekt kann es jedoch bestimmte Aktivitäten im Zusammenhang mit der Release-Vorbereitung geben, die nicht in allen Sprints im Projekt durchgeführt werden sollten. So kann es zum Beispiel geschäftlich Sinn machen, dass ein Projektteam einen vollständigen Satz teurer und zeitaufwändiger Leistungs- oder Lasttests oder einen speziellen Satz von End-to-End-Integrationstests nur kurz vor einem Release durchführt. Diese Aktivitäten liegen dann außerhalb der definierten Done-Kriterien für reguläre Sprints, und in solchen Fällen ist ein separater Release Vorbereitungs-Sprint erforderlich (siehe Abschnitt 13.3.7), um die zur Vorbereitung einer Freigabe erforderlichen Aufgaben abzuschließen.

13.3 Zusätzliche Hilfsmittel für große Projekte

13.3.1 Kommunikationsplan für das große Projekt

Der Kommunikationsplan für ein großes Projekt wird vom Chief Product Owner, dem Chief Scrum Master und dem Scrum Guidance Body erstellt, mit Beiträgen von anderen Product Ownern, Scrum Masters, Scrum Teams und anderen relevanten Personen.

Kommunikationspläne sind für große Projekte sehr wichtig, da Missverständnisse oder Mangel an Kommunikation im Projekt den gemeinsamen Anstrengungen abträglich sein und letztlich zum Scheitern des Projekts führen können. Ein Kommunikationsplan sollte Informationen darüber enthalten, wie die Kommunikation in allen Scrum Teams und mit den Business Stakeholdern erfolgen soll, einschließlich der zu verwendenden Kommunikationsmethoden, der Kommunikationskanäle oder -mechanismen für die Übermittlung wichtiger Informationen, der Verantwortlichkeiten für die Kommunikation, der Klassifizierung von und des Umgangs mit sensiblen Informationen, des Zeitplans für die Kommunikationsaktivitäten und der Prozesse zur Bewertung der Wirksamkeit der Kommunikation. Der Kommunikationsplan sollte auch den Zeitpunkt und die Häufigkeit der Scrum der Scrum Meetings und die Art und Weise, wie diese Meetings durchgeführt werden, enthalten.

Jedes Scrum Team kann auch seinen eigenen Kommunikationsplan haben (siehe Abschnitt 12.1.3.4), der die Aufzeichnungen festlegt, die erstellt und verteilt werden müssen, und wie diese Aufzeichnungen während des gesamten Projekts aufbewahrt werden. Der Plan sollte auch die Methoden enthalten, mit denen wichtige Projektinformationen an die Beteiligten weitergegeben werden und wer für die verschiedenen Kommunikationsaktivitäten verantwortlich ist.

Da es bei einem großen Projekt sehr schwierig sein kann, alle Beteiligten an einem Ort zu versammeln, kann die Verwendung eines Scrum Projekttools die effektive Kommunikation erleichtern.

13.3.2 Ressourcenplanung für das große Projekt*

Die Ressourcenplanung für große Projekte ist aufgrund der Komplexität der Zuteilung verschiedener Arten von Ressourcen an die zahlreichen parallel arbeitenden Scrum Teams unerlässlich. Es gibt konkurrierenden Bedarf für knappe Ressourcen, und der Chief Product Owner und die anderen Product Owner müssen planen, wie sie in der kürzest möglichen Zeit den größten Wert liefern können. Bei der Ressourcenplanung in einem großen Projekt sollten die verschiedenen Kosten für Ressourcen wie Mitarbeiter, Schulungen, Hard- und Software, externe Dienstleistungen und physische Räumlichkeiten berücksichtigt werden.

Der Chief Product Owner und andere Product Owner müssen sich möglicherweise mit externen Quellen abstimmen, um Ressourcen zu akquirieren und so den Mitarbeiterstab zu verstärken (es mag z. B. notwendig sein, externe Ressourcen zur Zusammenarbeit mit dem bestehenden Vollzeitteam zu akquirieren. Das erfolgt gegebenenfalls in Zusammenarbeit mit einem Partner-Management-Team, falls vorhanden. Bei der Akquirierung von externen Ressourcen müssen der Chief Product Owner und das Team die Unternehmensrichtlinien für den Umgang mit externen Ressourcen und Anbietern einhalten.

Bei großen Projekten muss der Chief Product Owner möglicherweise auch eine zusätzliche Ressourcenplanung in Betracht ziehen, um den Bedarf an spezialisierten Teams und die Notwendigkeit der Einrichtung von Umgebungen für zahlreiche parallel arbeitende Scrum Teams zu berücksichtigen. Der Chief Product Owner und die Product Owner arbeiten normalerweise mit den Scrum Mastern und Scrum Teams zusammen, um die speziellen für das große Projekt erforderlichen Fertigkeiten, die Anzahl der benötigten Ressourcen, die Scrum Teams, die spezialisierte Fähigkeiten benötigen, und die entsprechend Zuordnung zu ermitteln bzw. abzuschätzen.

13.3.3 Identifizierung der Umgebung(en)*

Der Chief Scrum Master arbeitet mit anderen Scrum Mastern, relevanten Product Ownern, Scrum Team-Mitgliedern, unterstützenden Diensten, dem Scrum Guidance Body und anderen Experten zusammen, um die erforderlichen und geeigneten Umgebungen zu identifizieren, die für die effektive Durchführung des großen Projekts benötigt werden. Dies sollte vorzugsweise einmal, während der Initiierungs-Phase, geschehen oder wenn es von den Teams und/oder dem SGB verlangt wird.

Bei einem großen Projekt ist es wichtig, die Anzahl und die Art der benötigten Umgebungen zu ermitteln, da zahlreiche Scrum Teams gleichzeitig an der Durchführung der Arbeit ihrer jeweiligen Sprints arbeiten werden und die Anforderungen an die Umgebungen kompliziert sein und stehen können. Einige Beispiele für Umgebungen sind Softwareentwicklungs- oder Testbereiche, physische Arbeitsbereiche oder Umgebungen mit spezieller Ausrüstung. Die Prozessgrenzen für jedes Scrum Team können sich ebenfalls auf die Umgebungen auswirken. Bei verteilten Teams, die in verschiedenen Zeitzonen arbeiten, kann es auch möglich sein, 24 Stunden am Tag zu testen und die Nutzung verschiedener Umgebungen zu maximieren. Daher ist es unerlässlich, einen Umgebungsplan zu erstellen, der die Testzeiten für jedes Team angibt. Bei Softwareprojekten kann der Umgebungsplan auch Informationen darüber enthalten, wie und von wem der Code in die einzelnen Umgebungen eingebracht wird.

13

13.3.4 Zuordnung des Priorisierten Produkt Backlogs*

In einem großen Projekt wird das priorisierte Produkt Backlog (mit den entsprechenden Epics und User Storys) vom Chief Product Owner und den anderen Product Ownern in ähnlicher Weise erstellt wie in typischen kleinen Scrum Projekten.

Die Erstellung von Epics und User Storys durch die Product Owner kann von mehreren Faktoren abhängen, wie z. B.:

- der Art und Weise, wie die Anforderungen von den Business Stakeholdernn gesammelt wurden
- Wissen und Erfahrung/Fertigkeiten der Product Owner

Der Chief Product Owner und die anderen Product Owner arbeiten zusammen, um das Priorisierte Produkt Backlog zu überprüfen und festzulegen, welche Epics und User Storys von den einzelnen Product Ownern bearbeitet werden sollen. Obwohl einige Epics und User Storys von einem bestimmten Product Owner erstellt werden, können sie einem anderen Product Owner zugewiesen werden, der sie als Teil des großen Scrum Projekts verwaltet und implementiert. Epics und User Storys müssen priorisiert werden, und einige bereits vorhandene Abschätzungen können verwendet werden, um ihre Zuordnung zu verschiedenen Product Ownern zu erleichtern.

Die Zuordnung von Epics und User Storys zu Product Ownern wird von verschiedenen Faktoren beeinflusst, wie z. B.:

- Der Product Owner, der die Epic/User Story erstellt hat (da er/sie oft für die Umsetzung verantwortlich sein wird)
- Der Product Owner, der mit einem Team zusammenarbeitet, das über die entsprechenden Fähigkeiten zur Fertigstellung des Epic/der User Story verfügt
- Der spezifische Kunde, Sponsor oder die Organisation, die mit der Epic/User Story verbunden ist (da eine frühere Beziehung zu einem bestimmten Product Owner bestehen könnte)
- Die Anzahl der Abhängigkeiten, die zu jeder Epic/User Story gehören (da die Gruppierung von User Storys mit vielen Abhängigkeiten unter einem Product Owner es den Teams ermöglichen könnte, relativ unabhängig voneinander zu arbeiten, ohne zu viel Zeit für die Koordination mit anderen Teams aufwenden zu müssen)

Die Zuordnung und Priorisierung von Epics oder User Storys an Product Owner kann im Laufe der Zeit durch Meetings und, wenn möglich, durch den Einsatz eines Scrum Projekttools erfolgen.

13.3.5 Scrum der Scrums Meeting*

Ein Scrum der Scrums Meeting ist ein wichtiges Element bei der Skalierung von Scrum auf große Projekte. Das Ziel der meisten Scrum der Scrums Meetings ist die Synchronisation der Teams während der Erstellung von zu Liefernden Ergebnissen, kann aber auch zum Austausch von Best Practices nach Sprint Retrospektions-Meetings und zur Planung zukünftiger Sprints genutzt werden. Die Häufigkeit von Scrum der Scrums Meetings ist projektspezifisch und hängt von der Projektgröße und -komplexität, den Abhängigkeiten zwischen verschiedenen Teams usw. ab. Wenn Epics oder User Storys in Sprints ohne allzu große Interaktion mit anderen Epics oder User Storys abgeschlossen werden können, sind diese Meetings möglicherweise nicht allzu oft erforderlich; sind die Abhängigkeiten hingegen hoch, kann eine höhere Frequenz von Scrum der Scrums Meetings erforderlich sein.

In der Regel nimmt ein Vertreter jedes Scrum Teams am Scrum der Scrums Meeting teil - in der Regel der Scrum Master -, aber es ist auch üblich, dass andere Mitglieder eines Scrum Teams an der Besprechung teilnehmen, wenn dies sinnvoll ist. Dieses Treffen wird in der Regel vom Chief Scrum Master moderiert und soll sich auf Bereiche der Koordination und Integration zwischen den verschiedenen Scrum Teams konzentrieren.

Dabei handelt es sich vorzugsweise um kurze Meetings, bei denen mindestens ein Vertreter jedes Scrum Teams (z.B. Scrum Master und/oder andere) zusammentrifft, um den Status der von den jeweiligen Teams geleisteten Arbeit auszutauschen, ähnlich wie beim täglichen Standup Meeting. Sie sind in der Regel nicht time-boxed, damit selbst bei sehr großen Projekten alle Vertreter ihre Informationen austauschen können. Das Scrum der Scrums Meeting wird in festgelegten Abständen oder auf Verlangen von Scrum Teams abgehalten, um den notwendigen Informationsaustausch zwischen den verschiedenen Scrum Teams zu erleichtern. Probleme, Abhängigkeiten und Risiken, die sich auf mehrere Scrum Teams auswirken, können genau beobachtet werden, was den verschiedenen Teams, die an einem großen Projekt arbeiten, hilft, ihre Arbeit besser zu koordinieren und zu integrieren. Es liegt in der Verantwortung des Chief Scrum Masters (oder eines anderen Scrum Masters, der die Scrum der Scrums Meetings moderiert), sicherzustellen, dass alle Vertreter ein Umfeld vorfinden, die einen offenen und ehrlichen Informationsaustausch ermöglicht, einschließlich Feedback an andere Teamvertreter. Bei größeren Projekten, an denen eine beträchtliche Anzahl von Teams beteiligt ist, können mehrere Ebenen von Scrum der Scrums Meetings einberufen werden, um den Status der einzelnen Teams zu teilen.

Jeder Scrum Team-Vertreter gibt der Reihe nach Updates aus seinem Team. Diese Aktualisierungen werden in der Regel in Form von Antworten auf die folgenden vier spezifischen Fragen gegeben.

1. Woran hat mein Team seit dem letzten Treffen gearbeitet?
2. Woran wird mein Team bis zum nächsten Treffen arbeiten?
3. Was haben andere Teams von unserem Team erwartet, was noch nicht erledigt wurde?
4. Was plant unser Team zu tun, das andere Teams beeinträchtigen könnte?

Die Antworten auf diese vier Fragen liefern Informationen, die es jedem Team ermöglichen, den Arbeitsstatus aller anderen Teams gut zu verstehen und festzustellen, ob es Probleme mit bevorstehenden Lieferungen gibt oder geben könnte.

13.3.6 Release Vorbereitungsmethoden*

Bei den Release Vorbereitungsmethoden handelt es sich um die Methoden, die zur Ausführung der im Release Vorbereitungsplan für die Vorbereitung der zu liefernden Produkte für die Auslieferung/Freigabe identifizierten Aufgaben verwendet werden. Diese Methoden können projektspezifisch sein, sind aber eher auf Programm- oder Portfolioebene gültig. Sie können durch den Scrum Guidance Body definiert werden.

13.3.7 Release Vorbereitungs-Sprint

Wenn es notwendig ist, bestimmte Aufgaben durchzuführen, um eine Freigabe vorzubereiten und um zu bestätigen, dass die Mindestanforderungen für die Freigabe erfüllt sind, werden diese Aufgaben in einem Release Vorbereitungs-Sprint durchgeführt. In einem Release Vorbereitungs-Sprint werden keine User Storys aus dem priorisierten Produkt Backlog entwickelt.

Stattdessen werden die im Release Vorbereitungsplan (siehe 13.2.7) festgelegten Aufgaben durchgeführt. Ein Release Vorbereitungs-Sprint wird nur einmal pro Release als erster Schritt im Prozess zu Liefernde Ergebnisse ausliefern durchgeführt. Die Länge eines Release Vorbereitungs-Sprints kann sich von der Länge anderer Sprints unterscheiden.

Wenn es einen Release Vorbereitungs-Sprint gibt, sind die zugehörigen Done-Kriterien in der Regel eindeutig und unterscheiden sich von den Done-Kriterien der User Storys für andere Sprints (die weiterhin erfüllt werden müssen). Die Done-Kriterien werden mit dem Ziel definiert, sicherzustellen, dass die Ergebnisse des Sprints "potenziell versandfähig" sind. Der Release Vorbereitungs-Sprint adressiert alle Aktivitäten, die nur einmal pro Release durchgeführt werden, basierend auf bewussten Geschäftsentscheidungen, die im Release Vorbereitungsplan begründet sind.

Ein Release Vorbereitungs-Sprint ist nicht obligatorisch, es sei denn, es gibt eine begründete geschäftliche Entscheidung, ihn in das Projekt einzubauen. Außerdem sind in der Regel nur die relevanten Teammitglieder an dem Release Vorbereitungs-Sprint beteiligt. Andere Scrum Teammitglieder, die nicht am Release Vorbereitungs-Sprint beteiligt sind, können mit der Arbeit an anderen regulären Sprints beginnen.

13.3.8 Scrum Projekttool

Da an großen Projekten viele Scrum Teams mit mehreren hundert oder mehr Personen beteiligt sind und die Teams auch verteilt sein können, kann es zu erheblicher Komplexität und Interaktionen zwischen den Teams kommen. Daher wäre es für die Teams von Vorteil, wenn sie Zugang zu einem strukturierten Scrum Projekttool oder einer Reihe von Tools hätten, die zur Automatisierung von Prozessen, zur Bewältigung der Komplexität, zur Weitergabe von Informationen (zwischen den Teams und an die Business Stakeholdern des Unternehmens), zur Erstellung von Berichten usw. verwendet werden können.

Einige spezifische Aufgaben, die das Scrum Projekttool bei einem großen Projekt übernehmen könnte, sind

- die Fähigkeit, Teams mit geeigneten Rollen zu bilden und die Fähigkeit, Rollen für große Projekte zu skalieren
- die Fähigkeit, ein priorisiertes Produkt Backlog für jedes Team zu erstellen und zu pflegen, einschließlich der Erstellung, Schätzung und Verwaltung von Epics, User Storys und Tasks
- die Fähigkeit zur Unterstützung anderer wichtiger Scrum Projektartefakte, wie Scrumboards, Sprint Backlogs, Besprechungspläne/Protokolle usw.
- die Fähigkeit, eine nahtlose und effektive Kommunikation zwischen allen Projektteammitgliedern zu ermöglichen
- die Fähigkeit zur Unterstützung verteilter Teams (siehe Abschnitt 2.5.3)
- die Fähigkeit, die Erstellung von Berichten und Metriken zu unterstützen, die von verschiedenen Scrum-Rollen benötigt werden
- die Fähigkeit, Empfehlungen oder Fachwissen aus dem Scrum Guidance Body zu erfassen und weiterzugeben (z.B. Erkenntnisse aus Sprints Retrospektionen, Best Practices, Scrum-bezogene Organisationsrichtlinien usw.)

14. Scrum für das Unternehmen Skalieren

Dieses Kapitel hebt zusätzliche Aspekte von Scrum hervor, die in Programmen und Portfolios zur Anwendung kommen. Die Skalierung von Scrum für das Unternehmen, wie sie im Scrum Body of Knowledge (SBOK® Guide) definiert ist, ist auf Folgendes anwendbar:

- Programme, Portfolios und/oder Projekte in jeder Branche
- Produkte, Dienstleistungen oder andere Ergebnisse, die an Business Stakeholdern geliefert werden sollen

Der Begriff "Programm" im SBOK® Guide bezieht sich auf eine Reihe zusammenhängender Projekte und/oder Unterprogramme, die auf koordinierte Weise verwaltet werden müssen, um Programmkomponenten zu produzieren und zu liefern. Der Begriff "Portfolio" bezieht sich auf eine Sammlung von Programmen und/oder Projekten innerhalb derselben Organisation, die direkt miteinander in Verbindung stehen oder auch unabhängig sein können. Sie müssen koordiniert verwaltet werden oder auch nicht, um die Portfolio-Ziele zu erreichen. Aus der Sicht des Scrum-Frameworks können Programme und Portfolios in ähnlicher Weise behandelt werden, aber auf unterschiedlichen Ebenen im Unternehmen, und erfordern daher möglicherweise ein unterschiedliches Maß an Koordination für die zugrunde liegenden Programme und/oder Projekte.

Dieses Kapitel befasst sich mit den Auswirkungen (auf Inputs, Hilfsmittel und Outputs), die ein Programm oder Portfolio auf die in den Kapiteln 8 bis 12 beschriebenen grundlegenden Scrum-Prozesse hat. In diesem Kapitel werden auch zusätzliche Prozesse vorgestellt, die ausschließlich für Programme und Portfolios gelten und auf Projektebene nicht relevant sind.

Um die bestmögliche Anwendung des Scrum-Frameworks zu ermöglichen, werden in diesem Kapitel Inputs, Hilfsmittel und Outputs für jeden Prozess entweder als "obligatorisch" oder "optional" bezeichnet. Inputs, Tools und Outputs, die mit einem Sternchen (*) gekennzeichnet sind, sind obligatorisch oder werden als kritisch für den Projekterfolg angesehen, während diejenigen ohne Sternchen optional sind.

Es wird empfohlen, dass Personen, die in die Anwendung des Scrum-Frameworks für das Unternehmen eingeführt werden, sich in erster Linie auf die obligatorischen Inputs, Hilfsmittel und Outputs konzentrieren, während Programm-/Portfolio-Product Owner, Chief Product Owner, Product Owner, Programm-/Portfolio-Scrum Master, Chief Scrum Master, Scrum Master und andere erfahrenere Scrum-Praktiker eine tiefergehende Kenntnis der Informationen in diesem gesamten Kapitel anstreben. Es ist auch wichtig zu wissen, dass, obwohl alle Prozesse im SBOK® Guide einzeln aufgeführt sind, sie nicht notwendigerweise sequenziell oder separat ausgeführt werden. Manchmal kann es sinnvoller sein, einige Prozesse parallel oder iterativ durchzuführen, je nach den spezifischen Anforderungen des jeweiligen Programms oder Portfolios.

Dieses Kapitel ist aus der Perspektive eines einzelnen Programm- oder Portfolio-Teams geschrieben, das die Aktivitäten mehrerer zugrunde liegender Scrum-Projekte und/oder -Programme koordiniert und priorisiert. Zusätzliche Informationen zur Anwendung von Scrum auf Projektebene finden Sie in den Kapiteln 2 bis 7, die Scrum-Prinzipien und Scrum-Aspekte behandeln.

Scrum für das Unternehmen vs. einzelnes Scrum-Projekt

Wenn man sich mit Scrum auf Unternehmensebene befasst, kann es mehrere hundert Scrum-Teams mit mehreren tausend Mitarbeitern geben, die in mehreren Projekten innerhalb von Programmen und/oder Portfolios im Unternehmen arbeiten. Die Anwendung von Scrum-Prozessen auf Programm- oder Portfolioebene hat bestimmte Auswirkungen auf die zugrunde liegenden Projekte. Im Allgemeinen werden die Scrum-Projekte weiterhin unter Verwendung der grundlegenden Scrum-Prozesse durchgeführt, die in den Kapiteln 8 bis 12 für typische kleine Projekte besprochen wurden; mit den zusätzlichen Inputs, Tools und Outputs, die in Kapitel 13 für große Projekte (mit mehreren Product Ownern und/oder Scrum Masters) beschrieben werden.

Die Auswirkungen von Programmen und Portfolios auf die in den Kapiteln 8 bis 12 beschriebenen Scrum-Prozesse auf Projektebene werden in Abschnitt 14.1 dieses Kapitels als zusätzliche Inputs, Hilfsmittel und Outputs skizziert. Die zusätzlichen Prozesse sowie Überlegungen, die nur auf der Programm- oder Portfolioebene relevant sind, werden in den Abschnitten 14.2 bis 14.8 behandelt.

Einige der Fragen, die sich auf der Programm- oder Portfolioebene stellen, ähneln denen, die sich in einem großen Scrum-Projekt stellen. Die Synchronisierung zwischen den Teams und die allgemeine Zusammenarbeit sind die größten Herausforderungen in einem großen Scrum-Projekt, und diese Herausforderungen bestehen auch auf der Programm- oder Portfolioebene. Die größten Herausforderungen für ein Programm oder Portfolio können jedoch auf der geschäftlichen Seite auftreten, da die geschäftlichen Prioritäten verschiedener Projekte sowohl untereinander als auch mit den Gesamtzielen des Programms oder Portfolios kollidieren können und aufeinander abgestimmt werden müssen.

Wie in einem großen Scrum-Projekt sind zusätzliche Inputs, Hilfsmittel und Outputs erforderlich, um den zusätzlichen Aufwand für die Priorisierung, Abstimmung und Koordination zu bewältigen. Einige Gründe für zusätzliche Inputs, Hilfsmittel und Outputs, die für Programme und/oder Portfolios benötigt werden, sind die Folgenden:

Product Owner

- Programm und Portfolio - Notwendigkeit der Abstimmung widersprüchlicher Geschäftsziele
- Programm - Notwendigkeit der Zusammenarbeit zwischen dem Product Owner des Programms und den Product Ownern der Projekte im Programm, z. B:
 - Verfeinerung des priorisierten Programm Backlogs,
 - Zusammenarbeit mit Business Stakeholdern, um Botschaften zu synchronisieren, und
 - Vermeidung von Doppelarbeit innerhalb des Programms (d.h. Synergie)
- Portfolio - Notwendigkeit der Zusammenarbeit zwischen dem Portfolio Product Owner, den Programm Product Ownern und den Product Ownern der Programme und Projekte im Portfolio, z. B:
 - Verfeinerung des priorisierten Portfolio-Backlogs,
 - die Zusammenarbeit mit den Business Stakeholdern, um Botschaften zu synchronisieren, und
 - Vermeidung von Doppelarbeit im Portfolio (d.h. Synergie)

Scrum-Master

- Programm und Portfolio - Notwendigkeit der Zusammenarbeit zwischen Scrum Masters bei der Beseitigung von Hindernissen
- Programm - Synchronisierung der Arbeit der Scrum Teams aus mehreren Projekten, falls erforderlich
- Portfolio - Synchronisierung der Arbeit der Scrum-Teams aus mehreren Programmen und Projekten, falls erforderlich

Scrum-Teams

- Programm und Portfolio - Notwendigkeit, Abhängigkeiten zwischen Scrum-Teams zu handhaben
- Programm und Portfolio - Notwendigkeit, gemeinsame Ressourcen und Ressourcenkonflikte zwischen Scrum Teams zu handhaben
- Programm und Portfolio-Notwendigkeit, bestimmte Richtlinien und Standards zu definieren, die von Scrum Teams für alle Projekte des Programms oder Portfolios eingehalten werden sollten (z.B. Sicherheitsstandards innerhalb der Organisation oder gesetzliche und/oder behördliche Vorschriften für bestimmte Branchen) - diese müssen möglicherweise vom Scrum Guidance Body dokumentiert werden.
- Programm und Portfolio - Erfordernis, eine Umgebung einzurichten und zu pflegen, die von mehreren Scrum-Teams genutzt werden kann

14.1 Auswirkungen von Programmen oder Portfolios auf die grundlegenden Scrum-Prozesse

Die Tabellen 14-1 bis 14-4 geben einen Überblick über die Auswirkungen von Programmen und Portfolios auf die grundlegenden Scrum-Prozesse für jede Projektphase.

14.1.1 Initiierung

Die zusätzlichen Inputs von der Programm-/Portfolioebene, die für die Initiierungsphase berücksichtigt werden müssen, sind wie folgt:

Prozess	Zusammenfassung der Auswirkungen eines Programms oder Portfolios
8.1 Projektvision erstellen	<p>Die Projektvision wird mit zusätzlichem Input aus dem Programm/Portfolio erstellt, zu dem das Projekt gehört. Darüber hinaus gibt es keine weiteren Änderungen an diesem Prozess.</p> <p>Zusätzlicher Input: Programm/Portfolio Product Owner Die Rollen des Programm/Portfolio Product Owners sind in den Abschnitten 3.7.4.1 und 3.7.4.2 beschrieben.</p> <p>Zusätzlicher Input: Programm-/Portfolio-Scrum-Master Die Rollen des Programm-/Portfolio-Scrum-Masters sind in den Abschnitten 3.7.4.3 und 3.7.4.4 beschrieben.</p> <p>Zusätzlicher Input: Programm-/Portfolio-Business-Business Stakeholdern Programm-/Portfolio-Business-Business Stakeholdern werden in Abschnitt 14.3.3.5 beschrieben. Sie haben Einfluss auf alle Projekte im Programm/Portfolio.</p> <p>Zusätzlicher Input: Priorisiertes Programm-/Portfolio-Backlog Das priorisierte Programm-/Portfolio-Backlog wird in Abschnitt 14.6.1.2 beschrieben. Das priorisierte Programm-/Portfolio-Backlog enthält Anforderungen für das Programm oder Portfolio, die sich auf die Projektvision auswirken können.</p>

Prozess	Zusammenfassung der Auswirkungen eines Programms oder Portfolios
8.2 Scrum Master und Business Stakeholdern identifizieren	<p>Der Scrum Master und der/die Business Stakeholdern des Projekts werden mit zusätzlichem Input aus dem Programm/Portfolio, zu dem das Projekt gehört, identifiziert. Darüber hinaus gibt es keine weiteren Änderungen an diesem Prozess.</p> <p>Zusätzlicher Input: Programm/Portfolio Product Owner Die Rollen des Programm-/Portfolio-Product Owners sind in den Abschnitten 3.7.4.1 und 3.7.4.2 beschrieben.</p> <p>Zusätzlicher Input: Programm-/Portfolio-Scrum-Master Die Rollen des Programm-/Portfolio-Scrum-Masters sind in den Abschnitten 3.7.4.3 und 3.7.4.4 beschrieben.</p> <p>Zusätzlicher Input: Programm-/Portfolio-Business-Business Stakeholdern Programm-/Portfolio-Business-Business Stakeholdern werden in Abschnitt 14.3.3.5 beschrieben. Sie beeinflussen alle Projekte im Programm oder Portfolio.</p>
Scrum Team bilden	Es gibt keine zusätzlichen Auswirkungen auf diesen Prozess für ein Projekt in einem Programm oder Portfolio.

Prozess	Zusammenfassung der Auswirkungen eines Programms oder Portfolios
8.4 Epic(s) entwickeln	<p>Die Epics werden mit zusätzlichem Input von dem Programm/Portfolio entwickelt, zu dem das Projekt gehört. Darüber hinaus gibt es keine weiteren Änderungen an diesem Prozess.</p> <p>Zusätzlicher Input: Priorisiertes Programm-/Portfolio-Backlog Das priorisierte Programm-/Portfolio-Backlog wird in Abschnitt 14.6.1.2 beschrieben. Das priorisierte Programm-/Portfoliobacklog enthält Anforderungen für das Programm oder Portfolio, die sich auf die Entwicklung von Epics auswirken können.</p> <p>Zusätzlicher Input: Programm-/Portfolio-Risiken Programm- und Portfoliorisiken werden in den Abschnitten 7.6.1 und 7.6.2 beschrieben. Risiken im Zusammenhang mit einem Programm oder Portfolio wirken sich auch auf die Projekte aus, die Teil des jeweiligen Programms oder Portfolios sind. Wenn bei der Risikobewertung des Programms oder Portfolios festgestellt wird, dass ein Risiko ein einzelnes Projekt beeinträchtigen könnte, müssen dem Product Owner und dem Scrum Team relevante Informationen über das Risiko mitgeteilt werden. Programm- und Portfoliorisiken werden zu Inputs für den Prozess Epic(s) entwickeln, und diese Risiken können sich auf die Durchführung dieses Prozesses auswirken.</p>
8.5 Priorisiertes Produkt Backlog erstellen	<p>Das priorisierte Produkt Backlog wird mit zusätzlichem Input aus dem priorisierten Programm/Portfolio Backlog erstellt. Darüber hinaus gibt es keine weiteren Änderungen an diesem Prozess.</p> <p>Zusätzlicher Input: Priorisiertes Programm-/Portfoliobacklog Das priorisierte Programm/Portfolio-Backlog wird in Abschnitt 14.6.1.2 beschrieben. Das priorisierte Programm-/Portfolio-Backlog enthält Anforderungen an das Programm/Portfolio, die sich auf die Erstellung des priorisierten Produkt-Backlogs auswirken können.</p>

Prozess	Zusammenfassung der Auswirkungen eines Programms oder Portfolios
8.6 Release-Planung durchführen	<p>Die Release Planung wird mit zusätzlichem Input aus dem Programm/Portfolio, zu dem das Projekt gehört, durchgeführt. Darüber hinaus gibt es keine weiteren Änderungen an diesem Prozess.</p> <p>Zusätzlicher Input: Programm/Portfolio Product Owner Die Rollen des Programm/Portfolio Product Owners sind in den Abschnitten 3.7.4.1 und 3.7.4.2 beschrieben.</p> <p>Zusätzlicher Input: Programm-/Portfolio-Scrum-Master Die Rollen des Programm-/Portfolio-Scrum-Masters sind in den Abschnitten 3.7.4.3 und 3.7.4.4 beschrieben.</p> <p>Zusätzlicher Input: Priorisiertes Programm-/Portfolio-Backlog Das priorisierte Programm-/Portfolio-Backlog wird in Abschnitt 14.6.1.2 beschrieben. Das priorisierte Programm/Portfolio-Backlog kann wichtige Koordinierungstermine und/oder Fristen für einige Anforderungen enthalten, die das Projekt einhalten muss.</p>

Tabelle 14-1: Auswirkungen eines Programms oder Portfolios auf die grundlegenden Scrum-Prozesse – Initiierungs-Phase

14.1.2 Planung und Abschätzung

Die Anwendung von Scrum auf Programm-/Portfolioebene hat keine Auswirkungen auf die Planungs- und Abschätzungsphase der jeweiligen Projekte im Programm/Portfolio.

14.1.3 Implementierung

Die zusätzlichen Inputs aus der Programm-/Portfolioebene, die in der Implementierungsphase berücksichtigt werden müssen, sind wie folgt:

Prozess	Zusammenfassung der Auswirkungen eines Programms oder Portfolios
10.1 Zu Liefernde Ergebnisse erzeugen	Die Erstellung der zu Liefernden Ergebnisse wird durch die Verwendung von Scrum auf der Programm- oder Portfolioebene nicht beeinflusst.

10.2 Tägliche Standup durchführen	Die Durchführung des täglichen Standups wird durch den Einsatz von Scrum auf der Programm- oder Portfolioebene nicht beeinflusst.
10.3 Priorisiertes Produkt Backlog verfeinern	<p>Bei der Verfeinerung des priorisierten Produkt-Backlogs müssen neue oder geänderte Anforderungen aus der Programm- oder Portfolio-Ebene entsprechend priorisiert und in das priorisierte Produkt-Backlog aufgenommen werden.</p> <p>Zusätzlicher Input: Programm-/Portfolio Product Owner Die Rollen des Programm-/Portfolio Product Owners sind in den Kapiteln 3.7.4.1 und 3.7.4.2 beschrieben. In diesem Prozess kommuniziert der Product Owner des Programms/Portfolios alle aktualisierten Anforderungen von der Programm- oder Portfolioebene an das Projekt.</p> <p>Zusätzlicher Input: Priorisiertes Programm-/Portfolio-Backlog Das priorisierte Programm-/Portfoliobacklog wird in Abschnitt 14.6.1.2 beschrieben. Alle Änderungen im priorisierten Programm-/Portfoliobacklog müssen in das priorisierte Produktbacklog des Projekts eingearbeitet werden.</p>

Tabelle 14-2: Auswirkungen eines Programms oder Portfolios auf die grundlegenden Scrum-Prozesse – Implementierungs-Phase

14.1.4 Review und Rückblick

Vertreter des Programms oder Portfolios können während der Sprint-Reviews oder der Retrospektions-Meetings Feedback geben. Darüber hinaus gibt es keine weiteren Änderungen in der Review- und Rückblick-Phase eines Projekts.

Prozess	Zusammenfassung der Auswirkungen eines Programms oder Portfolios
11.1 Sprint durchführen und validieren	<p>Vertreter aus dem Programm/Portfolio können Feedback geben. Darüber hinaus gibt es keine weiteren Änderungen an diesem Prozess.</p> <p>Zusätzlicher Input: Programm/Portfolio Product Owner</p> <p>Die Rollen des Programm/Portfolio Product Owners sind in den Abschnitten 3.7.4.1 und 3.7.4.2 beschrieben.</p> <p>Zusätzlicher Input: Programm-/Portfolio-Business-Business Stakeholdern</p> <p>Die Programm-/Portfolio-Business-Business Stakeholdern werden in Abschnitt 14.3.3.5 beschrieben.</p>
11.2 Sprint Rückblick halten	Der Prozess <i>Sprint Rückblick halten</i> wird durch die Verwendung von Scrum auf Programm- oder Portfolioebene nicht berührt.

Tabelle 14-3: Auswirkungen eines Programms oder Portfolios auf die grundlegenden Scrum-Prozesse Review- und Rückblick-Phase

14.1.5 Release

Die Anwendung von Scrum auf der Programm-/Portfolioebene kann sich auf die einzelnen Projekt-Releases auswirken, da es Abhängigkeiten zwischen den verschiedenen Projekt-Releases geben kann. Wenn zum Beispiel die Ergebnisse von zwei Projekten, A und B, idealerweise zusammen freigegeben werden sollten, sich aber die Ergebnisse von Projekt A verzögern, kann dies Auswirkungen auf die Freigabe der Ergebnisse von Projekt B haben, selbst wenn die Ergebnisse von Projekt B rechtzeitig fertiggestellt werden.

Prozess	Zusammenfassung der Auswirkungen eines Programms oder Portfolios
12.1 Zu Liefernde Ergebnisse ausliefern	<p>Die zu Liefernden Ergebnisse für ein Programm oder Portfolio werden auf die gleiche Weise erstellt wie für einzelne Projekte. Es kann jedoch Abhängigkeiten von zu Liefernden Ergebnissen aus anderen Projekten geben, die vom Product Owner des Programms/Portfolios oder dem Scrum Master des Programms/Portfolios koordiniert werden müssen.</p> <p>Aktualisierter Input: Programm/Portfolio Product Owner Beschrieben in Abschnitten 3.7.4.1 and 3.7.4.2.</p> <p>Aktualisierter Input: Programm/Portfolio Scrum Master Beschrieben in Abschnitten 3.7.4.3 and 3.7.4.4.</p>
12.2 Release Rückblick halten	Der Prozess <i>Release Rückblick halten</i> wird durch die Verwendung von Scrum auf Programm- oder Portfolioebene nicht berührt.

Tabelle 14-4: Auswirkungen eines Programms oder Portfolios auf die grundlegenden Scrum-Prozesse Release--Phase

14.2 Zusätzliche Prozesse zur Skalierung von Scrum für das Unternehmen (Programm/Portfolio)

Alle Scrum-Projekte, die Teil eines größeren Programms oder Portfolios sind, können Scrum-Prozesse anwenden, wie sie in den Kapiteln 8 bis 13 beschrieben sind, mit den zusätzlichen Auswirkungen, wie sie in Abschnitt 14.1 beschrieben sind.

Bei der Skalierung von Scrum für das Unternehmen können jedoch bestimmte zusätzliche Prozesse erforderlich sein, um die zusätzliche Komplexität von mehreren Hundert oder Tausend Personen, die an den zugehörigen Projekten arbeiten, und die zusätzlichen Koordinationsanforderungen auf Programm- oder Portfolioebene zu bewältigen. Alle diese Prozesse sind nicht notwendigerweise sequenziell und können parallel und iterativ angewendet werden, wie es das Unternehmen benötigt.

Abbildung 14-1 gibt einen Überblick über die zusätzlichen Prozesse, die für die Skalierung von Scrum für das Unternehmen erforderlich sind, und zwar:

14.3 Programm- oder Portfolio-Teams Bilden/Aktualisieren - In diesem Prozess werden zusätzliche Rollen zur Handhabung von Programmen und Portfolios geschaffen oder identifiziert. Zu diesen Rollen gehören der Programm Product Owner, der Portfolio Product Owner, der Programm Scrum Master, der Portfolio Scrum Master, die Business Stakeholder und die unterstützenden Dienste.

14.4 Programm- oder Portfoliokomponenten Erstellen/Aktualisieren - In diesem Prozess identifizieren und erstellen die Product Owner des Programms oder Portfolios, die Scrum Master und die Business Stakeholder die gemeinsamen Komponenten und Ressourcen, die für das Programm oder Portfolio erforderlich sind. Die minimalen Done-Kriterien werden definiert und alle anderen relevanten Business Stakeholder werden identifiziert. Abhängigkeiten zwischen Projekten werden adressiert, gemeinsame Hindernisse werden besprochen und bewährte Verfahren werden ausgetauscht. Manchmal werden auch Empfehlungen für Verbesserungen des Scrum Guidance Body ausgesprochen.

14.5 Scrum Guidance Body überprüfen und aktualisieren - In diesem Prozess werden die Empfehlungen des Scrum Guidance Body regelmäßig von den Mitgliedern des Scrum Guidance Body überprüft und bei Bedarf aktualisiert. Änderungen in der Zusammensetzung des Scrum Guidance Body werden ebenfalls behandelt. Das Hauptziel dieses Prozesses ist es, die Produktivität der Scrum-Projekte, -Programme und -Portfolios innerhalb der Organisation ständig zu beobachten und darauf hinzuarbeiten, sie zu verbessern.

14.6 Priorisiertes Programm- oder Portfoliobacklog erstellen/verfeinern - In diesem Prozess wird zunächst das Programm- oder Portfoliobacklog auf der Grundlage der Programm- oder Portfolioanforderungen erstellt. Das priorisierte Programm- oder Portfoliobacklog wird laufend überprüft, um Anforderungen, Risiken und Prioritäten hinzuzufügen oder zu aktualisieren.

14.7 Programm- oder Portfolio-Releases definieren/aktualisieren - In diesem Prozess werden die Programm- oder Portfolioreleases geplant, wobei alle Abhängigkeiten zwischen den Releases berücksichtigt werden. Die Programm- oder Portfolio-Release Planung wirkt sich auf die Release Planung auf Projektebene aus. Der Programm- oder Portfolio-Release Plan wird erstellt und sollte regelmäßig auf der Grundlage des Fortschritts der Projektergebnisse, neuer oder geänderter Anforderungen oder deren Prioritäten sowie anderer Faktoren aktualisiert werden.

14.8 Programm- oder Portfolio-Release Rückblick halten - In diesem Prozess kommen der Programm- oder Portfolio-Product Owner und die Business-Business Stakeholdern zusammen, um auf ein Programm- oder Portfolio-Release zurückzublicken und die daraus gewonnenen Erkenntnisse zu diskutieren und zu verinnerlichen. Häufig führen diese Erkenntnisse zu vereinbarten Verbesserungen, die in zukünftigen Releases umgesetzt werden sollen. Manchmal werden auch Verbesserungen des Scrum Guidance Body empfohlen.

14.3 Programm- oder Portfolio-Teams Bilden/Aktualisieren	14.4 Programm- oder Portfoliokomponenten Erstellen/Aktualisieren	14.5 Scrum Guidance Body überprüfen und aktualisieren
<p>INPUTS</p> <ol style="list-style-type: none"> 1. Unternehmens-Vision und Mission 2. Leitende Angestellte* 3. Organisatorische Ressourcenmatrix 4. Consultants <p>HILFSMITTEL</p> <ol style="list-style-type: none"> 1. Personalplanung des Unternehmens* 2. Stakeholder-Analyse <p>OUTPUTS</p> <ol style="list-style-type: none"> 1. Portfolio Product Owner* 2. Programm-Product Owner* 3. Portfolio-Scrum-Master* 4. Programm-Scrum-Master* 5. Business Stakeholder* 6. Unterstützende Dienste* 	<p>INPUTS</p> <ol style="list-style-type: none"> 1. Unternehmens-Vision und Mission 2. Portfolio Product Owner* 3. Portfolio-Scrum-Master* 4. Programm-Product Owner* 5. Programm-Scrum-Master* 6. Organisatorische Ressourcenmatrix 7. Empfehlungen des Scrum Guidance Body 8. Business Stakeholder* <p>HILFSMITTEL</p> <ol style="list-style-type: none"> 1. Kommunikationspläne* 2. Personalplanung des Unternehmens* 3. Stakeholder-Analyse 4. Scrum der Scrum (SoS) Meeting* 5. Scrum der Scrum der Scrum (SoSoS) Meeting 6. Kommunikationstechniken 7. Scrum Projekttool <p>OUTPUTS</p> <ol style="list-style-type: none"> 1. Minimale Done Kriterien* 2. Geteilte Ressourcen* 3. Identifizierte Business Stakeholder* 4. Aktualisierte Impediment Logs* 5. Aktualisierte Abhängigkeiten* 6. Product Owner Kollaborationsplan* 7. Scrum Master Kollaborationsplan * 8. Empfohlene Verbesserungen des Scrum Guidance Body 	<p>INPUTS</p> <ol style="list-style-type: none"> 1. Vorschriften* 2. Empfohlene Verbesserungen des Scrum Guidance Body* 3. Mitglieder des Scrum Guidance Body <p>HILFSMITTEL</p> <ol style="list-style-type: none"> 1. Auswahlkriterien für Mitglieder* 2. Scrum Guidance Body Meetings* 3. Performance Reports 4. Benchmarking <p>OUTPUTS</p> <ol style="list-style-type: none"> 1. Aktualisierte Empfehlungen des Scrum Guidance Body* 2. Erforderliche Eskalationen 3. Aktualisierte Besetzung des Scrum Guidance Body 4. Abgelehnte Aktualisierungen der Empfehlungen des Scrum Guidance Body

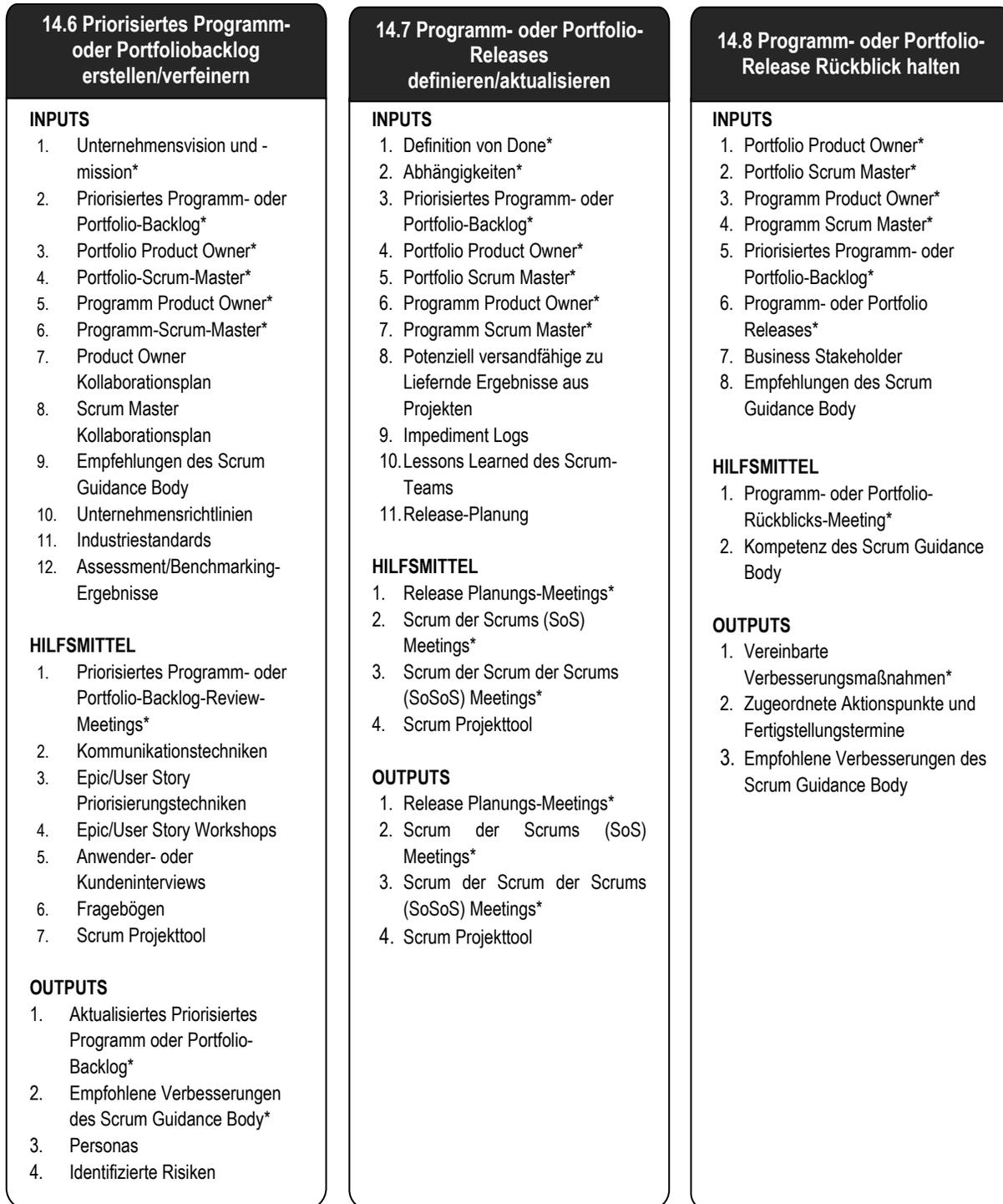


Abbildung 14-1: Skalierung von Scrum für das Unternehmen

14.3 Programm- oder Portfolio-Teams Bilden/Aktualisieren

Bevor Scrum-Prozesse in einer Unternehmensumgebung angewendet werden können, müssen Programm- oder Portfolio-Teams gebildet und/oder identifiziert werden. Einige wichtige Rollen sind der Programm Product Owner, der Portfolio Product Owner, der Programm Scrum Master, der Portfolio Scrum Master, Business Stakeholdern und unterstützende Dienste.

Es ist wichtig zu beachten, dass nicht alle diese Personen zu Beginn des Programms oder Portfolios identifiziert und zugeordnet werden müssen. Einige Schlüsselpersonen können bereits früh identifiziert werden, während andere je nach den spezifischen Anforderungen des Programms oder Portfolios im Laufe der Zeit zugeordnet werden können.

Die Ergebnisse des Prozesses *Programm- oder Portfolioteams bilden/aktualisieren* werden zu Inputs für den Prozess, nachdem das erste Team gebildet wurde. Sobald beispielsweise der Programm Product Owner oder der Programm Scrum Master identifiziert sind, werden diese Personen in alle zukünftigen Änderungen oder Aktualisierungen des Programmteams einbezogen.

Abbildung 14-2 zeigt alle Inputs, Hilfsmittel und Outputs für den Prozess *Programm- oder Portfolioteams bilden/aktualisieren*.

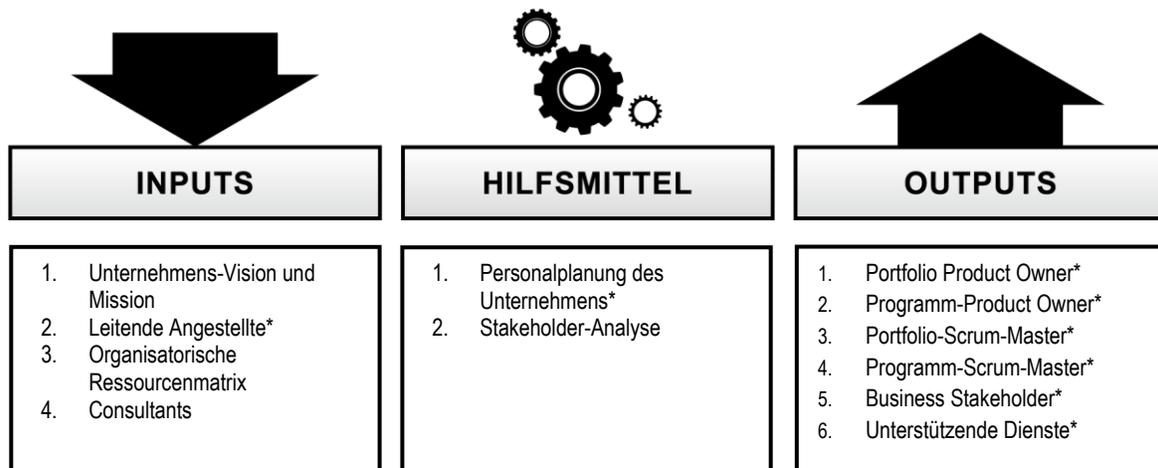


Abbildung 14-2: Programm oder Portfolio Teams bilden/aktualisieren—Inputs, Hilfsmittel, und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

14.3.1 Inputs

14.3.1.1 Unternehmens-Vision und Mission

Sowohl Unternehmens-Vision als auch -Mission sind für jedes Projekt wichtig, aber mehr noch für Programme und insbesondere auf der Portfolioebene. Programme und Portfolios sollten von der allgemeinen Mission und Vision des Unternehmens angetrieben werden, weil das die einheitliche Ausrichtung der Anstrengungen in der gesamten Organisation sicherstellt.

Das Verständnis der Unternehmensvision hilft den Portfolio- oder Programmteams, sich auf die Ziele der Organisation und das Zukunftspotenzial des Unternehmens zu konzentrieren. Die Programm- oder Portfolio Product Owner können sich von der Unternehmensvision leiten lassen, um die einzelnen Projektvisions-Erklärungen zu erstellen.

Die Unternehmensmission bietet einen Rahmen für die Formulierung der Strategien der Organisation und dient als Richtschnur für die allgemeine Entscheidungsfindung. Die Projekt-Visionserklärung muss so formuliert werden, dass ihre Erfüllung der Organisation hilft, ihre Mission zu erfüllen.

14.3.1.2 Leitende Angestellte*

Die Leitenden Angestellten sind hochrangige Vertreter der internen Organisation des Unternehmens, die von den Ergebnissen des Programms oder Portfolios profitieren oder für diese verantwortlich sind. Zu den leitenden Angestellten des Unternehmens können der Vorstandsvorsitzende (CEO), der Technologievorstand (CTO), der Finanzvorstand (CFO), Direktoren, Prokuristen und andere leitende Mitarbeiter aus verschiedenen Abteilungen des Unternehmens gehören.

14.3.1.3 Organisatorische Ressourcenmatrix

Die Matrix der Organisationsressourcen auf Programm- oder Portfolioebene sollte die Mitarbeiter in der Organisation enthalten, die verfügbar sind und über die Fähigkeiten verfügen, leitende Funktionen im Zusammenhang mit Scrum-Projekten zu übernehmen. Weitere Informationen über die Organisationsressourcenmatrix finden Sie in Abschnitt 8.2.1.5.

14.3.1.4 Consultants

Wenn nicht alle Fähigkeiten, die für den Aufbau und das Management von Scrum-Programmen oder -Portfolios innerhalb der Organisation erforderlich sind, intern im Unternehmen vorhanden sind, können externe Berater eingesetzt werden, um die leitenden Angestellten beim Aufbau der erforderlichen Programm- oder Portfolio-Teams zu unterstützen.

14.3.2 Hilfsmittel

14.3.2.1 Personalplanung des Unternehmens*

Die Personalplanung des Unternehmens enthält allgemeine Informationen darüber, wann bestimmte Mitarbeiter für verschiedene Projekte, Programme und Portfolios zur Verfügung stehen werden. Der Plan gibt auch Auskunft über die im Unternehmen vorhandenen Fähigkeiten und Fertigkeiten sowie über die Pläne zur Einstellung von Personal, das für künftige Aufgaben benötigt wird.

14.3.2.2 Stakeholdern-Analyse

Mit Hilfe verschiedener Stakeholdern-Analysetechniken können die Business Stakeholdern und alle anderen relevanten Business Stakeholdern, die auf Programm- und Portfolioebene betroffen sind, identifiziert und analysiert werden. Stakeholdern-Analysetechniken können auch eingesetzt werden, um die Interessen, die Beteiligung und die potenziellen Auswirkungen der einzelnen identifizierten Business Stakeholdern auf das Programm oder Portfolio zu bewerten. Eine Stakeholdern-Analyse ist auch hilfreich, um die Anforderungen an die Kommunikation und die Einbindung zu verstehen, die für den Erfolg des Programms oder Portfolios erforderlich sind.

14.3.3 Outputs

14.3.3.1 Portfolio Product Owner*

Siehe Abschnitt 3.7.4.2.

14.3.3.2 Programm-Product Owner*

Siehe Abschnitt 3.7.4.1.

14.3.3.3 Portfolio-Scrum-Master*

Siehe Abschnitt 3.7.4.4.

14.3.3.4 Programm-Scrum-Master*

Siehe Abschnitt 3.7.4.3.

14.3.3.5 Business Stakeholdern*

In diesem Prozess werden die Business Stakeholdern identifiziert, die eine Schlüsselrolle im Programm oder Portfolio spielen werden, einschließlich der Kunden, Anwender und Sponsoren des Programms oder Portfolios. Die Business Stakeholdern beeinflussen nicht nur das Programm oder Portfolio selbst, sondern auch die zugehörigen Projekte (innerhalb des Programms oder Portfolios) während ihrer gesamten Laufzeit. Programm- oder Portfolio-Business-Business Stakeholdern können auch dazu beitragen, die Projekt-/Programm-/Portfolio-Vision zu definieren und Hinweise zum Geschäftswert zu geben.

Die Programm-Business-Business Stakeholdern arbeiten mit den Portfolio-Business-Business Stakeholdern zusammen, um die Ausrichtung des Programms an den Zielen des Portfolios sicherzustellen. Programm- und/oder Portfolio-Business-Business Stakeholdern sind auch daran beteiligt, die Business-Business Stakeholdern für die einzelnen Projekte zu identifizieren und sicherzustellen, dass die Vision, die Ziele, die Ergebnisse und die Releases der zugehörigen Projekte innerhalb des Programms/Portfolios mit denen des Programms/Portfolios übereinstimmen.

Auf Portfolio-Ebene können zu den Business Stakeholdern auch Vorstandsmitglieder eines Unternehmens oder einer staatlichen Organisation gehören. Auf der Programmebene können zu den Business Stakeholdern auch leitende Angestellte und der oder die Sponsor(en) des Programms und der zugehörigen Projekte gehören.

14.3.3.6 Unterstützende Dienste*

Auf Programm- oder Portfolioebene sollten zu den unterstützenden Diensten Personen oder Gruppen gehören, die für die Bereiche Ausbildung, Logistik, Marketing, Finanzen, Infrastruktur, Architektur und andere unterstützende Dienstleistungen zuständig sind, die für den Erfolg des Programms oder Portfolios erforderlich sind. Einige dieser Personen können auch in Vollzeit für das Programm oder Portfolio tätig sein. Weitere Informationen über unterstützende Dienste finden Sie in Abschnitt 3.3.2.

14.4 Programm- oder Portfoliokomponenten Erstellen/Aktualisieren

In diesem Prozess identifizieren und erstellen die Product Owner des Programms oder Portfolios, die Scrum Master und die Business Stakeholdern die gemeinsamen Komponenten und Ressourcen, die für das Programm oder Portfolio erforderlich sind. Die minimalen Done Kriterien werden definiert und alle anderen relevanten Business Stakeholdern werden identifiziert. Abhängigkeiten zwischen Projekten werden adressiert, gemeinsame Hindernisse werden besprochen und Best Practices werden ausgetauscht. Manchmal werden auch Empfehlungen für Verbesserungen des Scrum Guidance Body ausgesprochen.

Es ist wichtig zu beachten, dass nicht alle Programm- oder Portfoliokomponenten zu Beginn erstellt werden müssen. Es handelt sich in der Regel um einen iterativen Prozess, bei dem einige wichtige Programm- oder Portfoliokomponenten bereits zu Beginn erstellt werden, während andere später erstellt oder aktualisiert werden können, wenn mehr Informationen zur Verfügung stehen.

Die Outputs des Prozesses "*Programm- oder Portfoliokomponenten erstellen/aktualisieren*" werden nach ihrer anfänglichen Erstellung zu Inputs für den Prozess. Sobald beispielsweise die minimalen Done Kriterien festgelegt sind, werden sie zu Inputs, wenn die Programm- oder Portfoliokomponenten aktualisiert werden.

Abbildung 14-3 zeigt alle Inputs, Hilfsmittel und Outputs für den Prozess "*Programm- oder Portfoliokomponenten erstellen/aktualisieren*".

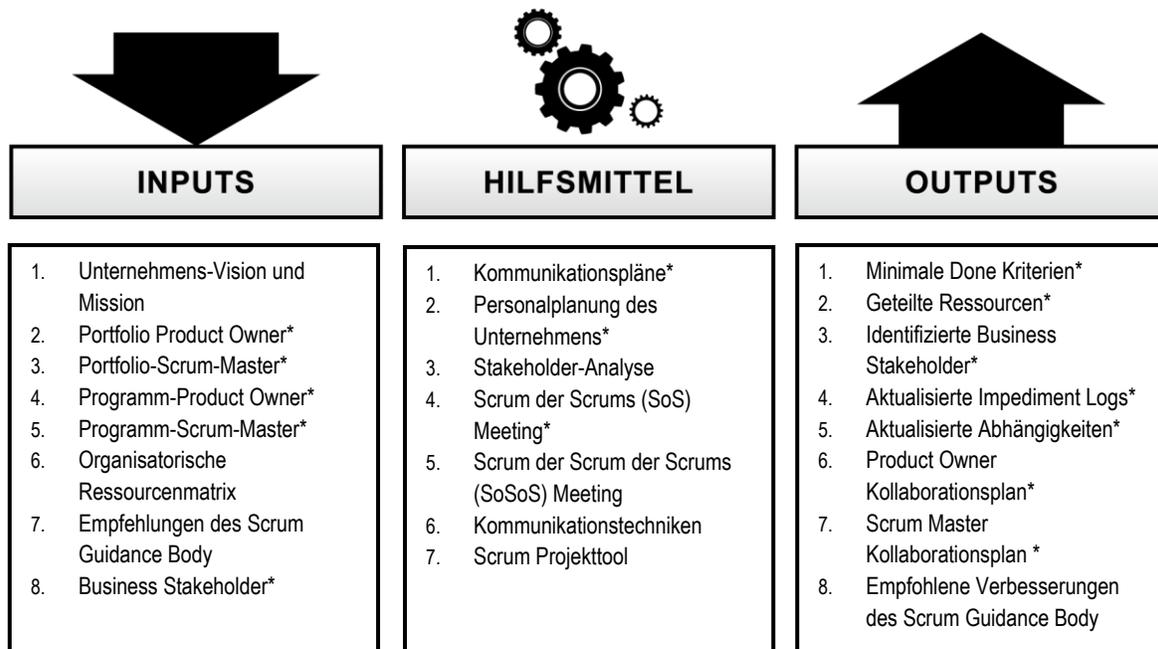


Abbildung 14-3: Programm oder Portfolio Komponenten erstellen/aktualisieren—Inputs, Hilfsmittel, und Outputs

Hinweis: Sternchen (*) bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess

14.4.1 Inputs

14.4.1.1 Unternehmens-Vision und Mission

Sowohl Unternehmens-Vision als auch -Mission sind für jedes Projekt wichtig, aber mehr noch für Programme und insbesondere auf der Portfolioebene. Programme und Portfolios sollten von der allgemeinen Mission und Vision des Unternehmens angetrieben werden, weil das die einheitliche Ausrichtung der Anstrengungen in der gesamten Organisation sicherstellt. Weitere Informationen über die Vision und die Mission des Unternehmens finden Sie in Abschnitt 14.3.1.1.

14.4.1.2 Portfolio Product Owner*

Siehe Abschnitt 3.7.4.2.

14.4.1.3 Portfolio-Scrum-Master*

Siehe Abschnitt 3.7.4.4.

14.4.1.4 Programm-Product Owner*

Siehe Abschnitt 3.7.4.1.

14.4.1.5 Programm-Scrum-Master*

Siehe Abschnitt 3.7.4.3.

14.4.1.6 Organisatorische Ressourcenmatrix

Siehe Abschnitt 8.2.1.5.

14.4.1.7 Empfehlungen des Scrum Guidance Body

Die Empfehlungen des Scrum Guidance Body sind besonders auf der Programm- und Portfolio-Ebene wichtig, da eine angemessene Anleitung für eine potenziell beträchtliche Anzahl von zusammenhängenden Projekten benötigt wird. Weitere Informationen zu den Empfehlungen des Scrum Guidance Body finden Sie in Abschnitt 8.1.1.7.

14.4.1.8 Business Stakeholdern*

Siehe Abschnitte 3.3.2 und 14.3.3.5.

14.4.2 Hilfsmittel

14.4.2.1 Kommunikationspläne*

Kommunikationspläne legen fest, wie Informationen an die Interessengruppen des Unternehmens sowie an die Programme, das Portfolio und die Organisation als Ganzes weitergegeben werden sollen. Es sollte auch festgelegt werden, wie und wann kommuniziert werden soll und welche Art der Kommunikation verwendet werden soll. Die Rollen des Portfolios bieten Anleitung und Input für den Kommunikationsplan für die zugehörigen Programme innerhalb des Portfolios. In ähnlicher Weise liefern die Rollen der Programme Anleitungen und Beiträge zum Kommunikationsplan für die Projekte innerhalb des Programms. Weitere Informationen zum Kommunikationsplan sind in Abschnitt 12.1.3.4 zu finden.

14.4.2.2 Personalplanung des Unternehmens*

Siehe Abschnitt 14.3.2.1.

14.4.2.3 Stakeholdern-Analyse

Siehe Abschnitt 14.3.2.2.

14.4.2.4 Scrum der Scrums (SoS) Meeting*

Der Zweck des Scrum der Scrums (SoS) Meetings ist so ähnlich wie bei großen Projekten. Auf der Programmebene treffen sich Vertreter jedes zugrunde liegenden Projekts im Programm in regelmäßigen Abständen zu Scrum der Scrums (SoS) Meetings. Weitere Informationen zu SoS-Meetings finden Sie in Abschnitt 13.3.5.

14.4.2.5 Scrum der Scrum der Scrums (SoSoS) Meeting

Auf der Programm- und vor allem auf der Portfolio-Ebene ist es sinnvoll, eine weitere Ebene von Meetings durchzuführen. In regelmäßigen Abständen oder nach Bedarf treffen sich Vertreter aus relevanten oder miteinander verbundenen Programmen und Projekten des Programms oder Portfolios. Anwesend sind Vertreter aus jedem der Scrum der Scrums Meetings. Diese zusätzliche Ebene von Meetings wird als Scrum der Scrum der Scrums (SoSoS) bezeichnet. Abbildung 14-4 veranschaulicht das Konzept der Scrum der Scrums (SoS) und Scrum der Scrum der Scrums (SoSoS) Meetings.

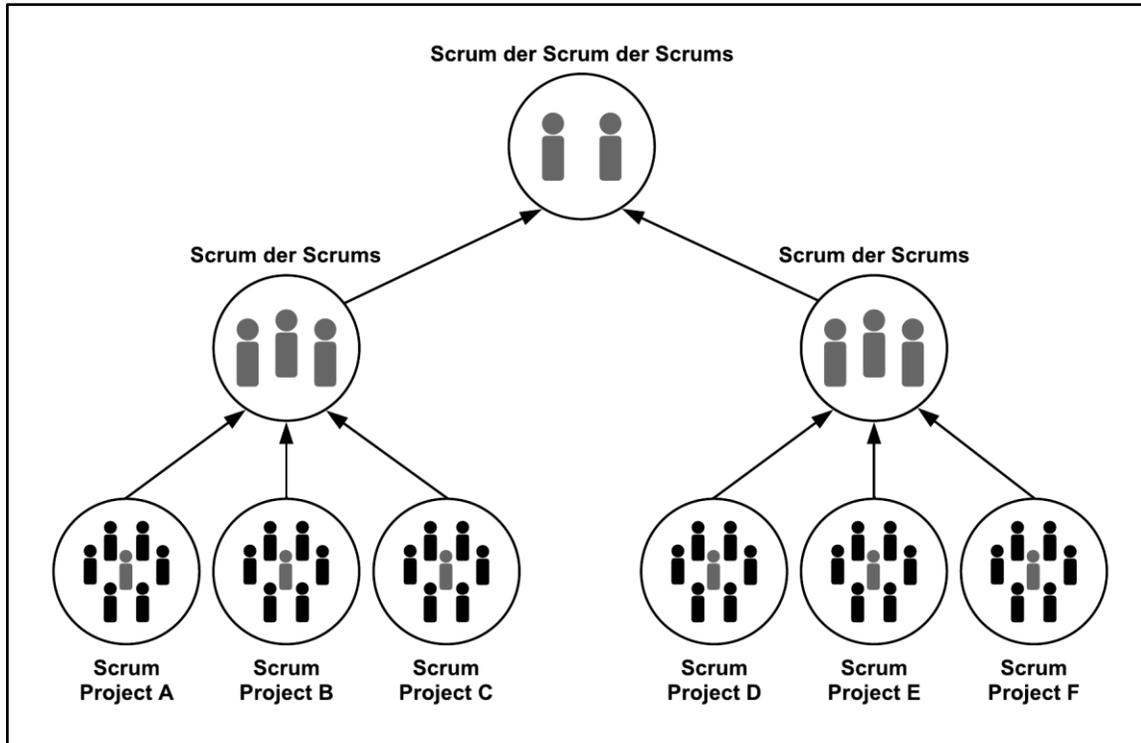


Abbildung 14-4: Scrum der Scrum der Scrums (SoSoS) Meeting

In diesem Beispiel gibt es sechs Scrum-Projekte, die gleichzeitig durchgeführt werden. Die Scrum-Projekte A, B und C sind Teil eines Programms, während die Scrum-Projekte D, E und F Teil eines anderen Programms sind. Es wird ein Scrum der Scrums Meeting abgehalten, um die gegenseitigen Abhängigkeiten in jedem der beiden Programme zu koordinieren. Anschließend kann ein Scrum der Scrum der Scrums Meeting durchgeführt werden, um die Abhängigkeiten zwischen den beiden Programmen des Portfolios zu koordinieren und zu managen.

14.4.2.6 Kommunikationstechniken

Auf Programm- oder Portfolioebene müssen die Kommunikationsmethoden und -techniken an eine größere Anzahl von Teilnehmern angepasst werden, auch aufgrund der Tatsache, dass sich nicht alle Teilnehmer am selben Arbeitsplatz befinden werden. Zusätzliche Überlegungen sollten zu Push- und Pull-Kommunikationsarten angestellt werden, z. B. zu Dashboards oder Berichten, die online veröffentlicht werden, damit die Beteiligten sie bei Bedarf einsehen können (Pull), oder die in regelmäßigen Abständen direkt verschickt werden (Push). Für die Kommunikation in größeren Gruppen können weitere Tools eingesetzt werden, z. B. Web-Meetings, E-Mail, Instant Messaging und Message Boards, wenn eine Kommunikation von Angesicht zu Angesicht nicht möglich ist. Der Einsatz eines Scrum-Projekttools kann auch die Kommunikation auf Programm- oder Portfolio-Ebene erleichtern. Weitere Informationen zu Kommunikationstechniken finden Sie in Abschnitt 10.3.2.2.

14.4.2.7 Scrum Projekttool

Auf Programm- oder Portfolioebene bietet das Scrum Projekttool die Möglichkeit der Skalierung, um die zusätzlichen Rollen, die Koordination, die Berichterstattung, die Kommunikation und andere relevante Anforderungen zu bewältigen. Weitere Informationen über das Scrum Projekttool finden Sie in den Abschnitten 2.5.3.1 und 13.3.8.

14.4.3 Outputs

14.4.3.1 Minimale Done Kriterien*

Die auf Portfolioebene definierten Minimalen Done Kriterien gelten für alle zugehörigen Programme und Projekte. Analog dazu gelten die Minimalen Done Kriterien auf Programmebene für alle zugehörigen Projekte. Der kaskadierende Satz von Done Kriterien stellt sicher, dass alle untergeordneten Done Kriterien die auf den höheren Ebenen festgelegten Minimalen Done Kriterien erfüllen. Der Scrum Guidance Body kann an der Definition der Minimalen Done Kriterien auf Portfolio-Ebene beteiligt sein. Für weitere Informationen zu den Minimalen Done Kriterien siehe Abschnitt 5.4.4.

14.4.3.2 Geteilte Ressourcen*

Siehe Abschnitt 13.2.4.

14.4.3.3 Identifizierte Business Stakeholdern*

Business Stakeholdern auf Portfolio- oder Programmebene sind ein Input für diesen Prozess. Weitere Business Stakeholdern werden im Rahmen dieses Prozesses identifiziert. Weitere Informationen über die auf Projektebene relevanten Geschäftsinteressenten sind in Abschnitt 8.2.3.2 zu finden.

14.4.3.4 Aktualisierte Impediment Logs*

Hindernisse, mit denen einzelne Projekte konfrontiert sind, können auch für andere Projekte innerhalb des Programms oder Portfolios relevant sein. Daher müssen die Impediment Logs auf Projektebene möglicherweise mit den anderen Projekten und/oder Programmen geteilt werden. Als Ergebnis der Scrum der Scrums (SoS) oder Scrum der Scrum der Scrums (SoSoS) Meetings kann es erforderlich sein, die Impediment Logs auf Projektebene zu aktualisieren. Es kann aber auch Impediment Logs auf Programm- oder Portfolio-Ebene geben. Weitere Informationen über das Impediment Log finden Sie in Abschnitt 10.1.1.4.

14.4.3.5 Aktualisierte Abhängigkeiten*

Es kann Abhängigkeiten zwischen miteinander verbundenen Projekten und sogar zwischen Programmen innerhalb des Unternehmens geben, die ermittelt werden müssen. Folglich sollte es eine Koordination zwischen den zugehörigen Projekten geben, um diese Abhängigkeiten zu managen.

Beispiele für Abhängigkeiten können sein:

- Gemeinsame Veröffentlichungstermine für miteinander verbundene Projekte
- Abhängigkeiten zwischen Releases
- Abhängigkeiten in Bezug auf miteinander verbundene Features

Durch die Koordinierung von Programm- oder Portfoliokomponenten kann es erforderlich werden, die bekannten Abhängigkeiten durch neue Abhängigkeiten oder Änderungen an bestehenden Abhängigkeiten zu aktualisieren. So kann es zum Beispiel Abhängigkeiten zwischen Projekten innerhalb eines Programms oder Portfolios geben. Bei zwei Projekten A und B in einem Programm kann es sein, dass diese beiden Projekte denselben Release-Termin haben müssen, oder aber, dass Projekt A erst nach der Freigabe von Projekt B freigegeben werden kann. Weitere Informationen über Abhängigkeiten auf Projektebene finden Sie in den Abschnitten 8.5.2.6 und 9.4.2.3.

14.4.3.6 Product Owner Kollaborationsplan*

Siehe Abschnitt 13.2.2.

14.4.3.7 Scrum Masters/Scrum Teams Kollaborationsplan *

Siehe Abschnitt 13.2.3.

14.4.3.8 Empfohlene Verbesserungen des Scrum Guidance Body

Als Ergebnis des Prozesses Programm- oder Portfoliokomponenten erstellen/aktualisieren können Vorschläge oder Feedback für mögliche Verbesserungen der Dokumentation des Scrum Guidance Body gegeben werden. Diese Verbesserungsvorschläge werden vom Scrum Guidance Body diskutiert und angenommen oder abgelehnt (siehe Abschnitt 14.5, Scrum Guidance Body überprüfen und aktualisieren). Wenn den Vorschlägen zugestimmt wird, werden sie als Aktualisierungen in die Dokumentation des Scrum Guidance Body aufgenommen.

14.5 Scrum Guidance Body überprüfen und aktualisieren

In diesem Prozess werden die empfohlenen Verbesserungen des Scrum Guidance Body regelmäßig von den Mitgliedern des Scrum Guidance Body überprüft und bei Bedarf aktualisiert. Änderungen in der Zusammensetzung des Scrum Guidance Body werden ebenfalls in diesem Prozess berücksichtigt. Das Hauptziel besteht darin, die Produktivität der Scrum-Projekte, -Programme und -Portfolios innerhalb der Organisation ständig zu beobachten und darauf hinzuarbeiten, sie zu verbessern.

Es ist wichtig zu beachten, dass der Prozess *Scrum Guidance Body überprüfen und aktualisieren* typischerweise ein iterativer Prozess ist, da Scrum-Projekte kontinuierlich Retrospektions-Meetings durchführen, in denen regelmäßig Verbesserungsmöglichkeiten identifiziert und an die Programm- und Portfolioebene weitergeleitet werden. Auf der Programm- oder Portfolioebene prüft der Scrum Guidance Body die Inputs aus den Retrospektions-Meetings auf Programm- und Portfolioebene, identifiziert Verbesserungsmöglichkeiten und hilft bei der Verbreitung von guten Praktiken im gesamten Unternehmen.

Abbildung 14-5 zeigt alle Inputs, Hilfsmittel und Outputs für den Prozess *Scrum Guidance Body überprüfen und aktualisieren*.

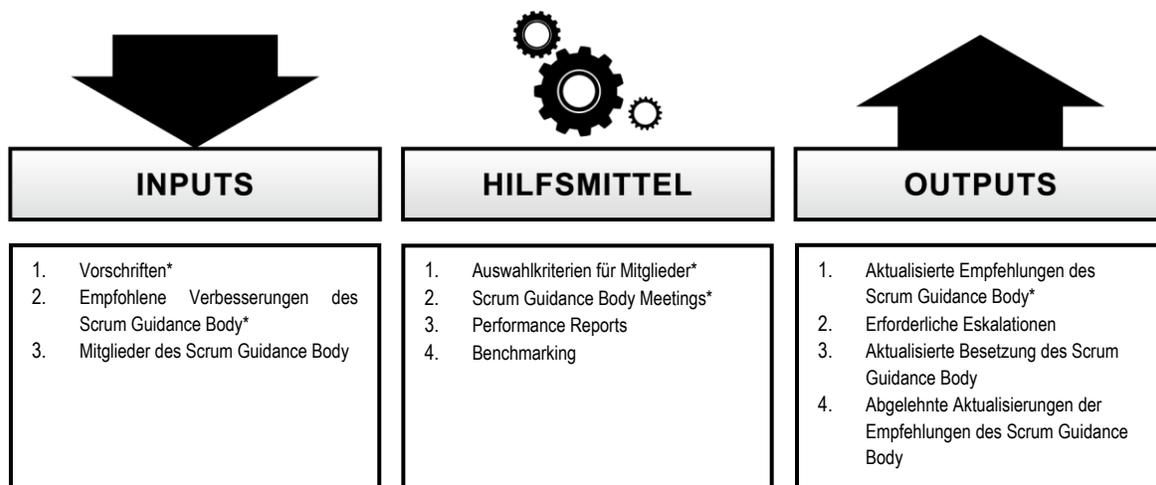


Abbildung 14-5: Scrum Guidance Body überprüfen und aktualisieren —Inputs, Hilfsmittel, und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

14.5.1 Inputs

14.5.1.1 Vorschriften*

Zu den Vorschriften gehören solche des Bundes, der Länder, der Städte und Gemeinden oder der Branche, an die sich das Programm oder Portfolio halten muss. User Stories, die erstellt werden, um staatliche Vorschriften innerhalb eines bestimmten Zeitraums zu erfüllen, werden in das Portfolio- oder Programm-Produkt-Backlog aufgenommen. Gelegentlich müssen die Empfehlungen des Scrum Guidance Body aktualisiert werden, um neue Vorschriften zu berücksichtigen.

14.5.1.2 Empfohlene Verbesserungen des Scrum Guidance Body*

Als Ergebnis von Scrum Retrospektions-Meetings und anderen Prozessen können Vorschläge und Feedback zur Überarbeitung oder Verbesserung der Richtlinien, Vorlagen und anderer Dokumentation des Scrum Guidance Body gemacht werden. Wenn der Scrum Guidance Body mit den Vorschlägen oder dem Feedback einverstanden ist, werden die entsprechenden Änderungen als Aktualisierungen in das Material des Scrum Guidance Body aufgenommen und als Empfehlungen an die Projekt-, Programm- und Portfolio-Teams weitergegeben.

14.5.1.3 Mitglieder des Scrum Guidance Body

Zu den Mitgliedern des Scrum Guidance Body (SGB) können Scrum-Experten, Scrum-Coaches, externe Berater, ausgewählte Scrum Master, Product Owner und Teammitglieder (auf allen Ebenen) gehören. Die Anzahl der Mitglieder des SGB sollte jedoch begrenzt sein, um sicherzustellen, dass es relevant bleibt und keinen präskriptiven Charakter annimmt.

14.5.2 Hilfsmittel

14.5.2.1 Auswahlkriterien für Mitglieder*

Die Auswahlkriterien für Mitglieder werden erstellt, um die Mitglieder des Scrum Guidance Body, ihre Rollen und Verantwortlichkeiten, die Anzahl der Mitglieder und ihre erforderlichen Fähigkeiten und Kenntnisse zu definieren. Jede Organisation kann ihre eigenen Auswahlkriterien für die Mitglieder des Scrum Guidance Body haben; es wird jedoch empfohlen, dass jedes Mitglied über Scrum Know-how verfügt und effektiv zum Scrum Guidance Body beitragen kann.

14.5.2.2 Scrum Guidance Body Meetings*

Der Scrum Guidance Body trifft sich in regelmäßigen Abständen, um einen möglichen Aktualisierungsbedarf der Empfehlungen des Scrum Guidance Body zu besprechen (z.B. Verbesserungsvorschläge aus Retrospektions-Meetings und anderen Prozessen, aktualisierte Vorschriften, die in die Dokumentation eingearbeitet werden müssen, usw.). Die Häufigkeit dieser Treffen wird vom Scrum Guidance Body auf der Grundlage der spezifischen Bedürfnisse des Unternehmens festgelegt.

14.5.2.3 Performance Reports

Es können Berichte über die Performance von Scrum-Projekten, -Programmen und -Portfolios verfügbar sein. Solche Performance-Reports können Informationen über die Geschwindigkeit eines Teams, die gelieferte Funktionalität, den Status der Fertigstellung und so weiter enthalten. Diese Informationen können vom Scrum Guidance Body bei der Ermittlung von Verbesserungsmöglichkeiten berücksichtigt werden.

14.5.2.4 Benchmarking

Benchmarking ist der Prozess des Vergleichs der Geschäftsprozesse und Leistungs-Metriken einer Organisation mit denen führender Unternehmen der gleichen oder anderer Branchen. Ein Unternehmen sollte seine eigenen Praktiken regelmäßig mit denen erfolgreicher Organisationen (um mit der Konkurrenz Schritt zu halten) sowie mit aktuellen und aufkommenden Branchenstandards und -praktiken vergleichen.

14.5.3 Outputs

14.5.3.1 Aktualisierte Empfehlungen des Scrum Guidance Body*

Nach Prüfung und Berücksichtigung der Verbesserungsvorschläge des Scrum Guidance Body, der Performance Reports und der Benchmarking-Daten können Änderungen an der bestehenden Dokumentation erforderlich sein. Alle genehmigten Änderungen führen zu einer Aktualisierung des Scrum Guidance Body-Materials und werden als Empfehlungen für die laufenden oder zukünftigen Scrum-Projekte, -Programme und -Portfolios zur Verfügung gestellt.

14.5.3.2 Erforderliche Eskalationen

Das Scrum Guidance Body kann feststellen, dass einige Unternehmensrichtlinien es den Teams nicht erlauben, den maximalen Nutzen aus der Anwendung von Scrum zu ziehen. In solchen Fällen sollte eine Eskalation ausgelöst werden, um eine Genehmigung für eine Richtlinienänderung oder eine andere Änderung zu erhalten.

14.5.3.3 Aktualisierte Besetzung des Scrum Guidance Body

Als Ergebnis der Bewertung der Zusammensetzung des Scrum Guidance Body können neue Mitglieder hinzukommen und bisherige Mitglieder den Scrum Guidance Body verlassen.

14.5.3.4 Abgelehnte Aktualisierungen der Empfehlungen des Scrum Guidance Body

Empfohlene Verbesserungen des Scrum Guidance Body werden nicht immer angenommen. Wenn eine empfohlene Verbesserung von den Mitgliedern des Scrum Guidance Body abgelehnt wird, wird eine Erläuterung der Gründe für die Ablehnung als Feedback an die betroffenen Parteien gegeben.

14.6 Priorisiertes Programm- oder Portfoliobacklog erstellen/verfeinern

In diesem Prozess wird zunächst der Programm- oder Portfolio-Backlog auf der Grundlage der Programm- oder Portfolio-Anforderungen erstellt. Das priorisierte Programm- oder Portfoliobacklog wird laufend mit neuen oder aktualisierten Anforderungen, Risiken und Prioritäten aktualisiert und gepflegt.

Alle Ergebnisse des Prozesses "Priorisiertes Programm- oder Portfoliobacklog erstellen/verfeinern" werden nach ihrer anfänglichen Erstellung zu Inputs für den Prozess. Das priorisierte Programm- oder Portfoliobacklog wird zum Beispiel zum ersten Mal während dieses Prozesses erstellt, aber es wird zu einem obligatorischen Input für nachfolgende Verfeinerungen/Aktualisierungen.

Abbildung 14-6 zeigt alle Inputs, Hilfsmittel und Outputs für den Prozess Priorisiertes Programm oder Portfolio-Backlog erstellen/verfeinern.

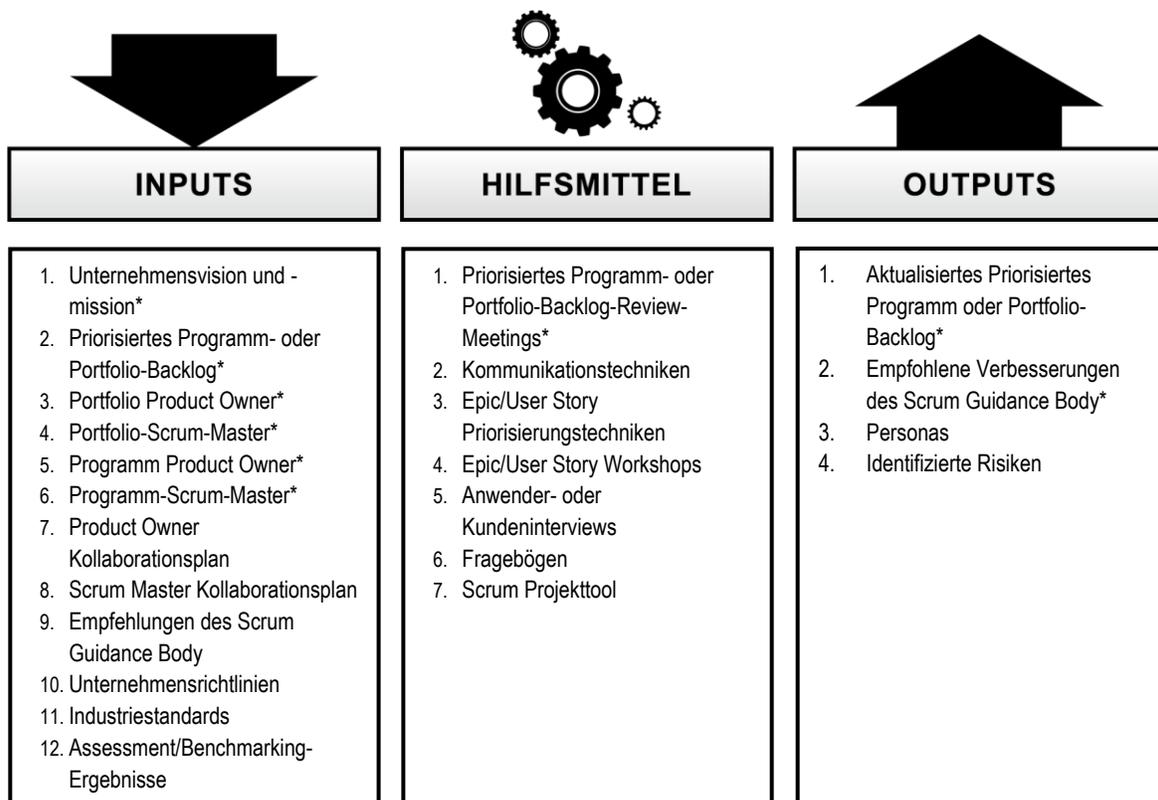


Abbildung 14-6: Priorisiertes Programm- oder Portfoliobacklog erstellen/verfeinern —Inputs, Hilfsmittel, und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

14.6.1 Inputs

14.6.1.1 Unternehmensvision und -mission*

Siehe Abschnitt 14.3.1.1.

14.6.1.2 Priorisiertes Programm- oder Portfolio-Backlog*

Der Product Owner des Programms oder Portfolios entwickelt das priorisierte Programm- oder Portfolio-Backlog, das eine priorisierte Liste von high-level Geschäfts- und Projektanforderungen enthält, die vorzugsweise in Form von großen Epics auf Programm- oder Portfolioebene geschrieben werden. Diese werden später von den Product Ownern der einzelnen Projekte verfeinert, wenn sie die Product Backlogs für ihre eigenen Projekte erstellen und priorisieren. Diese priorisierten Product Backlogs haben viel kleinere, aber detailliertere User Stories, die von den einzelnen Scrum Teams abgeschätzt und zugesagt werden können. Diese Details werden zwischen verschiedenen Projekten ausgetauscht, um unnötige Doppelarbeit zu vermeiden.

Das priorisierte Programm Backlog spielt auf der Programmebene eine sehr ähnliche Rolle wie das priorisierte Produkt Backlog auf der Projektebene. Es erfasst die Anforderungen für das Programm und deren Prioritäten. Es gibt jedoch ein paar Unterschiede. Die Erstellung der jeweiligen zu Liefernden Ergebnisse und deren Abnahme werden innerhalb der Projekte des Programms abgewickelt. Auf der Programmebene können für jedes Product Backlog Element/User Story Done- oder Abnahmekriterien definiert werden. Die Teams müssen sich an diese Kriterien halten, es können aber bei Bedarf weitere Kriterien hinzugefügt werden.

Die Länge eines Sprints ist projektspezifisch und kann daher von Projekt zu Projekt innerhalb eines Programms variieren. Auch die Geschwindigkeit variiert von Team zu Team. Daher ist es nicht notwendig, auf Programmebene sehr granulare User Stories zu erstellen. Die Verfeinerung der Epics auf Programmebene geht nur so weit, dass sichergestellt ist, dass jedes einzelne Epic klar verstanden wird und dass konkrete Abnahmekriterien für das Programm definiert werden können.

Das Priorisierte Portfolio-Backlog spielt auf der Portfolio-Ebene die gleiche Rolle wie das Priorisierte Programm-Backlog auf der Programm-Ebene. Die Elemente des priorisierten Portfolio-Backlogs dienen als Input für die verschiedenen priorisierten Programm-Backlogs und schließlich für die priorisierten Produkt-Backlogs der entsprechenden Projekte. Wie beim priorisierten Programmbacklog wird auf dieser Ebene, wenn überhaupt, nur eine minimale Verfeinerung der Epics vorgenommen, da die Verfeinerung innerhalb der zugehörigen Projekte auf der Ebene der einzelnen priorisierten Product Backlogs erfolgt.

Das priorisierte Programm- oder Portfolio-Backlog wird vom Programm- oder Portfolio Product Owner kontinuierlich weiterentwickelt, um sicherzustellen, dass neue Geschäftsanforderungen hinzugefügt und bestehende Anforderungen ordnungsgemäß dokumentiert und priorisiert werden. Auf diese Weise wird sichergestellt, dass die für die Erreichung der Ziele des Portfolios oder Programms wertvollsten Anforderungen hoch priorisiert werden und die übrigen Anforderungen eine niedrigere Priorität erhalten.

Das priorisierte Programm- oder Portfoliobacklog bietet ein umfassendes Bild aller Projekte, die Teil des Programms oder Portfolios sind. Er kann daher wichtige Hinweise auf die Projektziele, den Umfang, die Zielsetzungen und den erwarteten geschäftlichen Nutzen liefern.

14.6.1.3 Portfolio Product Owner*

Der Portfolio Product Owner ist für die Erstellung und Verfeinerung des priorisierten Portfolio-Backlogs verantwortlich. Weitere Informationen über die Rolle des Portfolio Product Owners finden Sie in Abschnitt 3.7.4.2.

14.6.1.4 Portfolio-Scrum-Master*

Auf der Portfolioebene spielt der Portfolio Scrum Master eine ähnliche Rolle wie der Programm Scrum Master für ein Programm. Er oder sie ist ein Moderator, löst Probleme und beseitigt Hindernisse auf Portfolioebene. Weitere Informationen über die Rolle des Portfolio Scrum Masters finden Sie in Abschnitt 3.7.4.4.

14.6.1.4 Programm Product Owner*

Auf der Programmebene ist der Programm Product Owner für die Erstellung und Verfeinerung des priorisierten Programm Produkt Backlogs verantwortlich und treibt diese voran. Weitere Informationen zur Rolle des Programm Product Owners finden Sie in Abschnitt 3.7.4.1.

14.6.1.5 Programm-Scrum-Master*

Auf der Programmebene spielt der Programm Scrum Master eine ähnliche Rolle wie der Scrum Master in einem Projekt. Er oder sie ist ein Vermittler, löst Probleme und beseitigt Hindernisse auf der Programmebene. Weitere Informationen über die Rolle des Programm Scrum Masters finden Sie in Abschnitt 3.7.4.3.

14.6.1.6 Product Owner Kollaborationsplan

Siehe Abschnitt 13.2.2.

14.6.1.7 Scrum Masters/Scrum Teams Kollaborationsplan

Siehe Abschnitt 13.2.3.

14.6.1.8 Empfehlungen des Scrum Guidance Body

Bei der Erstellung und Verfeinerung des priorisierten Programm- oder Portfolio-Backlogs bieten die Empfehlungen des Scrum Guidance Body Best Practices, die auf Programm- oder Portfolio-Ebene berücksichtigt werden sollten. Weitere Informationen zu den Empfehlungen des Scrum Guidance Body finden Sie in den Abschnitten 8.1.1.7 und 10.3.1.11.

14.6.1.9 Unternehmensrichtlinien

Unternehmensrichtlinien sind eine Reihe von Prinzipien, Regeln und Richtlinien, die von einer Organisation formuliert oder übernommen wurden. Eine Änderung der Unternehmensrichtlinien kann sich auf bestehende Epics oder User Storys auswirken, die auf der Grundlage bestehender Richtlinien erstellt wurden.

14.6.2 Industriestandards

Neue Industriestandards oder Änderungen an bestehenden Standards müssen implementiert werden, um ein funktionsfähiges Produkt oder eine nützliche Dienstleistung zu erhalten. Daher müssen User Storys, die sich auf die Erfüllung dieser Standards beziehen, in das priorisierte Programm oder Portfolio Backlog aufgenommen und entsprechend priorisiert werden. Manchmal müssen auch die Empfehlungen des Scrum Guidance Body geändert werden, um neue oder geänderte Industriestandards zu berücksichtigen.

14.6.2.1 Assessment/Benchmarking-Ergebnisse

In erster Linie werden die Ergebnisse von Assessments oder Benchmarking eine Aktualisierung der Empfehlungen des Scrum Guidance Body für Best Practices erforderlich machen. Die Ergebnisse können auch dazu beitragen, einen Mindeststandard für die Erstellung eines Produkts oder einer Dienstleistung festzulegen und können zu Änderungen an den Done Kriterien führen. Manchmal können neue Assessment- oder Benchmarking-Ergebnisse auch den Anstoß für einen Programm oder Portfolio Product Owner geben, neue Epics zu entwickeln, um zusätzliche oder aktualisierte Best Practices zu implementieren.

14.6.3 Hilfsmittel

14.6.3.1 Priorisiertes Programm- oder Portfolio-Backlog-Review-Meetings*

Die Teilnahme an den Programm oder Portfolio Backlog Review Meetings unterscheidet sich deutlich von der Teilnahme an den Product Backlog Review Meetings auf Projektebene. Scrum Teams nehmen an den entsprechenden Meetings auf der Projektebene teil. Auf der Programm- oder Portfolioebene ist jedes Projekt innerhalb des Programms oder jedes Programm und/oder einzelne Projekte innerhalb des Portfolios vertreten. Um die Sitzungen zu straffen, wird jedoch im Allgemeinen empfohlen, nur einen oder einige wenige Vertreter aus jedem Projekt oder Programm auf Programm- oder Portfolioebene teilnehmen zu lassen. Weitere Informationen finden Sie in den entsprechenden Abschnitten 6.5.1.2 und 10.3.2.1.

14.6.3.2 Kommunikationstechniken*

Siehe Abschnitt 10.3.2.2

14.6.3.3 Epic/User Story Priorisierungstechniken

Auf der Programm- oder Portfolioebene gibt es normalerweise eine geringere Anzahl von Anforderungen/Epics/User Storys als auf der Projektebene. Außerdem sind diese Anforderungen auf einem high-level Niveau und die Priorisierung wird in erster Linie von den geschäftlichen Anforderungen (wie von den Business Stakeholdern, dem Portfolio Product Owner und dem Programm Product Owner bestimmt). Weitere Informationen zu Priorisierungsmethoden finden Sie in Abschnitt 8.5.2.1.

14.6.3.4 Epic/User Story Workshops

Im Vergleich zu Projekten zielen User Story Workshops für Programme und Portfolios nur auf die Erstellung von Higher-Level Epics/User Stories ab, so dass es zu diesem Zeitpunkt deutlich weniger Epics/User Stories geben wird. Die Meetings sind jedoch immer noch wertvoll, da an ihnen Repräsentanten der Projekte innerhalb des Programms oder der Programme innerhalb des Portfolios teilnehmen und diese Personen relevante Informationen an ihre jeweiligen Teams weitergeben können. Diese Workshops werden in der Regel vom Programm- oder Portfolio-Scrum-Master koordiniert. Dadurch wird sichergestellt, dass die Anforderungen im gesamten Programm oder Portfolio gut definiert und verstanden werden. Weitere Informationen zu User Story Workshops finden Sie in Abschnitt 8.4.2.2.

14.6.3.5 Anwender- oder Kundeninterviews

Siehe Abschnitt 8.4.2.4.

14.6.3.6 Fragebögen

Siehe Abschnitt 8.4.2.5.

14.6.3.7 Scrum Projekttool

Ein angemessen konzipiertes Scrum Projekttool bietet eine leicht verständliche Ansicht des priorisierten Programm- oder Portfolio-Backlogs und hilft dem Programm- oder Portfolio-Produktverantwortlichen, die Anforderungen/Epics/User Stories zu sehen und zu priorisieren. Weitere Informationen über das Scrum Projekttool finden Sie in den Abschnitten 2.5.3.1 und 13.3.8.

14.6.4 Outputs

14.6.4.1 Aktualisiertes Priorisiertes Programm oder Portfolio-Backlog*

Das Priorisierte Programm- oder Portfolio-Backlog kann mit neuen oder aktualisierten Epics/User Storys, mit Arbeiten im Zusammenhang mit neuen Change Requests oder identifizierten Risiken und/oder zwecks Re-Priorisierung bestehender Epics/User Storys aktualisiert werden.

Die Verfeinerung des Programm- oder Portfolio-Backlogs kann zu einer Entscheidung führen, ein neues Projekt zu initiieren (z.B. zur Erstellung eines Referenzsystems oder einer gemeinsamen Web-Schnittstelle, die von allen Projekten genutzt werden soll).

14.6.4.2 Empfohlene Verbesserungen des Scrum Guidance Body*

Als Ergebnis des Prozesses zur Erstellung/Verfeinerung des Programm- oder Portfolio-Backlogs können Vorschläge oder Rückmeldungen für mögliche Verbesserungen der Dokumentation des Scrum Guidance Body gemacht werden. Diese Verbesserungsvorschläge werden vom Scrum Guidance Body diskutiert und angenommen oder abgelehnt. Wenn einer der Vorschläge angenommen wird, wird er als Update in die Dokumentation des Scrum Guidance Body aufgenommen. Weitere Informationen finden Sie im Prozess "Scrum Guidance Body Überprüfen und Aktualisieren".

14.6.4.3 Personas

Siehe Abschnitt 8.4.3.2.

14.6.4.4 Identifizierte Risiken

Risiken, die sich auf ein Programm oder Portfolio beziehen, wirken sich auch auf die Projekte aus, die Teil des jeweiligen Programms oder Portfolios sind. Wenn während der Risikobewertung des Programms oder Portfolios festgestellt wird, dass ein Risiko ein untergeordnetes Programm oder Projekt beeinträchtigen kann, müssen relevante Informationen über dieses Risiko an den jeweiligen Product Owner und das Scrum Team übermittelt werden. Programm- und Portfoliorisiken werden zu Inputs für den Prozess Epics entwickeln für das/die entsprechende(n) Projekt(e) und können sich insgesamt auf die Durchführung dieses Prozesses auswirken. Weitere Informationen zur Identifizierung von Projektrisiken finden Sie in den Abschnitten 7.4.1 und 8.4.3.4. Programm- und Portfoliorisiken werden in Abschnitt 7.6 behandelt.

14.7 Programm- oder Portfolio-Releases definieren/aktualisieren

In diesem Prozess werden die Programm- oder Portfolio-Releases geplant, wobei alle Abhängigkeiten zwischen den Releases berücksichtigt werden. Die Programm- oder Portfolio Release Planung hat Auswirkungen auf die Release Planung auf Projektebene. Der Programm- oder Portfolio-Release Plan wird erstellt und sollte regelmäßig auf der Grundlage des Fortschritts der zu liefernden Projektergebnisse, neuer oder geänderter Anforderungen oder deren Prioritäten sowie anderer Faktoren überprüft werden.

Das Release Planungsmeeting dient dazu, die bestehenden Releases zu begutachten und neue Releases zu planen.

Die Ergebnisse des Prozesses *Programm- oder Portfolioreleases definieren/aktualisieren* werden nach ihrer ursprünglichen Erstellung zu Inputs für den Prozess. Beispielsweise können während dieses Prozesses zum ersten Mal Programm- oder Portfolioreleases erstellt werden, die jedoch zu obligatorischen Eingaben für künftige Programm- oder Portfolioreleases werden.

Abbildung 14-7 zeigt alle Inputs, Hilfsmittel und Outputs für den Prozess *Programm- oder Portfolioreleases definieren/aktualisieren*.

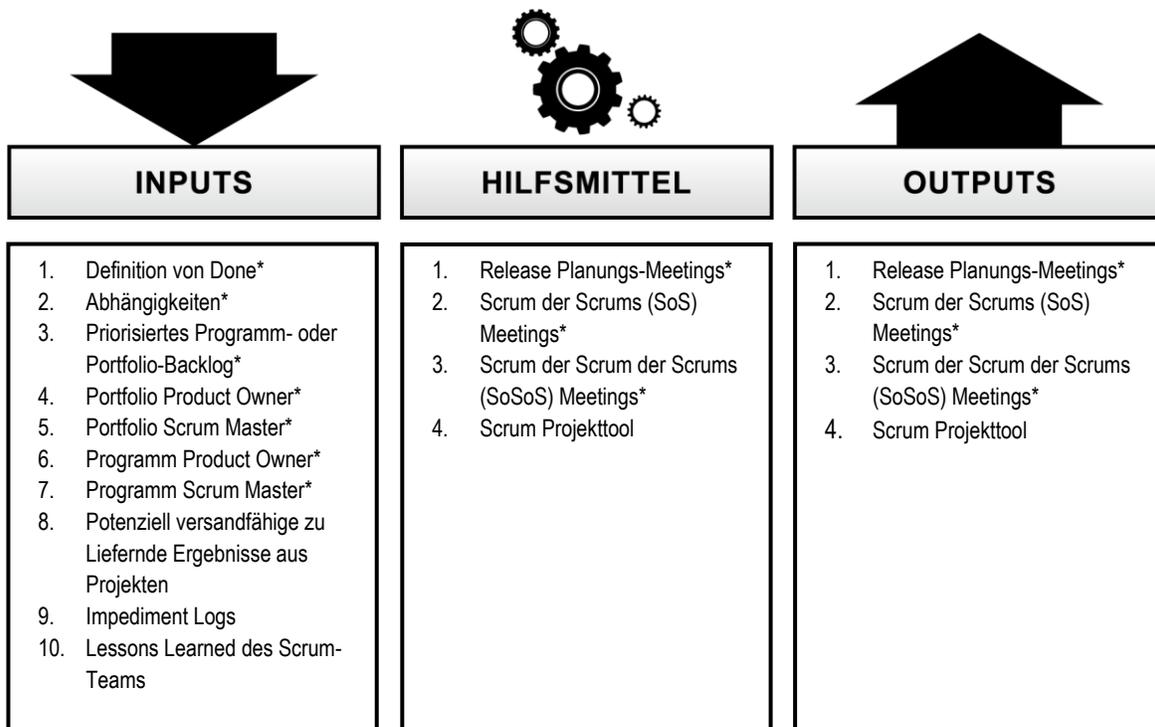


Abbildung 14-7: Priorisiertes Programm- oder Portfolioreleases definieren/aktualisieren —Inputs, Hilfsmittel, und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

14.7.1 Inputs

14.7.1.1 Definition von Done*

Die auf Programm- oder Portfolioebene definierte Definition von Done (oder Done Kriterien) kann als minimale Done Kriterien für Projekte im gesamten Unternehmen verwendet werden. Weitere Informationen zu den Done Kriterien finden Sie in den Abschnitten 5.4.3 und 8.5.3.2.

14.7.1.2 Abhängigkeiten*

Siehe Abschnitte 8.5.2.6 und 8.5.3.5.

14.7.1.3 Priorisiertes Programm- oder Portfolio-Backlog*

Siehe Abschnitt 14.6.1.2.

14.7.1.4 Portfolio Product Owner*

Siehe Abschnitt 3.7.4.2.

14.7.1.5 Portfolio Scrum Master*

Siehe Abschnitt 3.7.4.4.

14.7.1.6 Programm Product Owner*

Siehe Abschnitt 3.7.4.1.

14.7.1.7 Programm Scrum Master*

Siehe Abschnitt 3.7.4.3.

14.7.1.8 Potenziell versandfähige zu Liefernde Ergebnisse aus Projekten

Potenziell versandfähige zu Liefernde Ergebnisse aus Projekten sind wertvolle Inputs für die Koordination auf Programm- oder Portfolioebene. Am Ende eines jeden Sprints in einem Projekt werden Produktinkremente oder zu Liefernde Ergebnisse fertiggestellt. Die in diesen Inkrementen enthaltenen User Storys erfüllen sowohl die Done Kriterien als auch ihre jeweiligen Abnahmekriterien.

14.7.1.9 Impediment Logs

Siehe Abschnitte 10.1.1.4 und 14.4.3.4.

14.7.2 Lessons Learned des Scrum-Teams

Siehe Abschnitt 11.2.3.5.

14.7.3 Release-Planung

Diese Planung ist zwar nur vorläufig und unterliegt Änderungen, aber sie ist von entscheidender Bedeutung, um zu beurteilen, ob die jeweiligen Projekte die geforderten Termine voraussichtlich einhalten werden, und sie ist insbesondere im Hinblick auf Abhängigkeiten von entscheidender Bedeutung. Weitere Informationen über die Release-Planung auf Projektebene sind in Abschnitt 8.6.3.1 zu finden.

14.7.4 Hilfsmittel

14.7.4.1 Release Planungs-Meetings*

Release Planungs-Meetings finden zwischen dem Portfolio Product Owner, dem Programm Product Owner, dem Portfolio Scrum Master, dem Programm Scrum Master und anderen relevanten Business Stakeholdern aus Geschäfts- und Projektteams statt, um sicherzustellen, dass alle Programm- und Portfolio-Releases richtig geplant werden. Die Programm- und Portfolio-Releases liefern wiederum wertvolle Informationen für die Planung der Releases auf Projektebene.

14.7.4.2 Scrum der Scrums (SoS) Meetings*

Siehe Abschnitt 13.3.5.

14.7.4.3 Scrum der Scrum der Scrums (SoSoS) Meetings*

Siehe Abschnitt 14.4.2.5.

14.7.4.4 Scrum Projekttool

Ein Scrum Projekttool hilft dem Team, geplante Programm- oder Portfolio-Releases anzuschauen, entsprechende Änderungen einzubringen, falls erforderlich, und zusätzliche Releases zu planen. Weitere Informationen über das Scrum Projekttool sind in den Abschnitten 2.5.3.1 und 13.3.8 zu finden.

14.7.5 Outputs

14.7.5.1 Programm- oder Portfolio Releases*

Ein Programm- oder Portfoliorelease umfasst die Releases für alle zugehörigen Projekte des Programms oder Portfolios. Es gibt in der Regel zwei Arten von Programm- oder Portfolioreleases:

1. Alle Leistungen für die zugehörigen Projekte sind abgeschlossen, werden aber bereitgehalten, um gemäß den Vorgaben des Programms oder Portfolios freigegeben zu werden. Es kann einen bestimmten Freigabetermin geben, an dem das gesamte Programm oder Portfolio für den Endkunden freigegeben wird.
2. Zu Liefernde Ergebnisse werden auf Projektebene freigegeben, sobald sie fertig sind, z.B. bei DevOps, das eine kontinuierliche Entwicklung, Implementierung und Verteilung ermöglicht.

14.7.5.2 Programm- oder Portfolio-Release-Plan*

Ein Programm- oder Portfolio-Release-Plan enthält Zieldaten für verschiedene auf Programm- oder Portfolio-Ebene geplante Releases.

14.7.5.3 Aktualisiertes priorisiertes Programm- oder Portfolio-Backlog

Der priorisierte Programm- oder Portfoliobacklog wird aktualisiert, wenn bestehende Releases überprüft und vorgeschlagene Änderungen oder neue Releases hinzugefügt werden. Die Aktualisierung von Programm- oder Portfolio-Releases kann sich auch auf die Priorisierung von Epics/User Storys im priorisierten Programm- oder Portfolio-Backlog auswirken. Weitere Informationen über das priorisierte Programm- oder Portfolio-Backlog finden Sie in Abschnitt 14.6.1.2.

14.7.5.4 Aktualisierte Abhängigkeiten

Abhängigkeiten können aufgrund von Diskussionen in den Release Planungs-Meetings oder -Sessions aktualisiert werden.

14.8 Programm- oder Portfolio-Release Rückblick halten

In diesem Prozess kommen der Programm- oder Portfolio Product Owner und die Business Stakeholdern zusammen, um einen Rückblick auf ein Programm- oder Portfolio-Release zu halten und die gewonnenen Erkenntnisse zu diskutieren und zu verinnerlichen. Oft führen diese Erkenntnisse zu vereinbarten Verbesserungen, die in zukünftigen Releases umgesetzt werden sollen. Manchmal werden auch Verbesserungen des Scrum Guidance Body empfohlen. Diese Meetings können nach jedem Programm- oder Portfolio-Release angesetzt werden.

Abbildung 14-8 zeigt alle Inputs, Hilfsmittel und Outputs für den Prozess Programm oder Portfolio Release Rückblick halten.

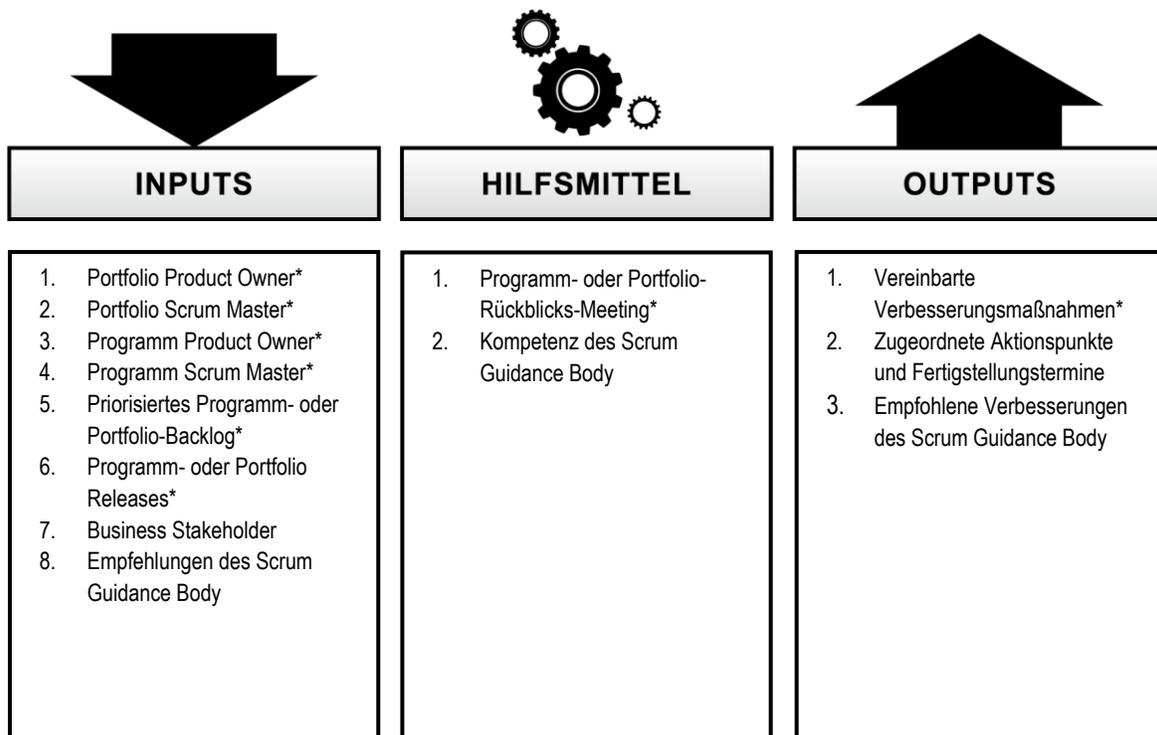


Abbildung 14-8: Programm- oder Portfolio-Release Rückblick halten —Inputs, Hilfsmittel, und Outputs

Hinweis: Sternchen () bezeichnen einen verbindlichen Input, Hilfsmittel oder Output für den entsprechenden Prozess*

14.8.1 Inputs

14.8.1.1 Portfolio Product Owner*

Siehe Abschnitt 3.7.4.2.

14.8.1.2 Portfolio Scrum Master*

Siehe Abschnitt 3.7.4.4.

14.8.1.3 Programm Product Owner*

Siehe Abschnitt 3.7.4.1.

14.8.1.4 Programm Scrum Master*

Siehe Abschnitt 3.7.4.3.

14.8.1.5 Priorisiertes Programm- oder Portfolio-Backlog*

Siehe Abschnitt 14.6.1.2.

14.8.1.6 Programm- oder Portfolio Releases*

Siehe Abschnitt 14.7.3.1.

14.8.1.7 Business Stakeholdern

Beschrieben in den Abschnitten 3.3.2 und 14.3.3.5.

14.8.1.8 Empfehlungen des Scrum Guidance Body

Während eines Rückblicks auf Programm- oder Portfolio-Releases liefern die Empfehlungen des Scrum Guidance Body einschlägige Best Practices, einschließlich Informationen zu administrativen Verfahren, Audits, Bewertungen und Kriterien für Projektübergänge. Dies ist vergleichbar mit der Rolle, die die Empfehlungen des Scrum Guidance Body bei den Rückblicken auf Projektebene spielen (beschrieben in Abschnitt 12.2.1.5).

14.8.2 Hilfsmittel

14.8.2.1 Programm- oder Portfolio-Rückblicks-Meeting*

Das Programm- oder Portfolio-Rückblicks-Meeting ähnelt dem Release-Rückblicks-Meeting auf Projektebene (siehe Abschnitt 12.2.2.1). Der Hauptunterschied besteht darin, dass die Programm- und Portfolio-Rückblicks-Meetings viel seltener stattfinden als die Release-Rückblicks-Meetings und dass sie den Programm/Portfolio Product Owner, den Programm/Portfolio Scrum Master und die Business Stakeholdern für das Programm oder Portfolio einschließen.

14.8.2.2 Kompetenz des Scrum Guidance Body

Siehe Abschnitt 8.4.2.7.

14.8.3 Outputs

14.8.3.1 Vereinbarte Verbesserungsmaßnahmen*

Siehe Abschnitt 11.2.3.1.

14.8.3.2 Zugeordnete Aktionspunkte und Fertigstellungstermine

Siehe Abschnitt 11.2.3.2

14.8.3.2 Empfohlene Verbesserungen des Scrum Guidance Body

Als Ergebnis des Prozesses Programm oder Portfolio Release Rückblick halten können Vorschläge oder Rückmeldungen für mögliche Verbesserungen der Dokumentation des Scrum Guidance Body gemacht werden. Diese Verbesserungsvorschläge werden vom Scrum Guidance Body diskutiert und angenommen oder abgelehnt. Wenn einer der Vorschläge angenommen wird, wird er als Update in die Dokumentation des Scrum Guidance Body aufgenommen. Weitere Informationen finden Sie im Prozess "Scrum Guidance Body überprüfen undaktualisieren".

ANHANG A. EIN AGILE ÜBERBLICK

A.1 Einleitung

Der Zweck dieses Anhangs ist den Leser mit dem Konzept und der Etymologie von Agile vertraut zu machen und die verschiedenen, branchenübergreifend angewendet Agile Methoden vorzustellen.

Dieses Kapitel ist in folgende Abschnitte eingeteilt:

A.2 Überblick—Die Definition von und die Faktoren hinter dem Aufstieg von agilen Methoden werden besprochen.

A.3 Das Agile Manifest—Das Agile Manifest, seine Prinzipien und die Interdependenz-Erklärung werden besprochen, um den historischen Kontext für Agile zu erläutern.

A.4 Agile Methoden—Ein kurzer Überblick verschiedener Agile Methoden, unter Anderem:

- Lean Kanban
- Extreme Programming
- Crystal Methoden
- Dynamische Systementwicklungsmethode
- Featuregetriebene Entwicklung (Feature Driven Development)
- Testgetriebene Entwicklung (Test Driven Development)
- Adaptive Software Entwicklung (Adaptive Software Development)
- Agile Unified Process
- Domain Getriebener Entwurf (Domain Driven Design)

A.2 Überblick

Der Duden definiert den Begriff 'agil', als 'von grosser Beweglichkeit zeugend; regsam und wendig'. Agil bedeutet flink sein. Agil ist schnell reagieren zu können. Agilität im Management ist ein wichtiges Ziel. Agile Projektmanagement ist ein anpassungsfähiges Projektmanagement.

Um es mit den Worten von Jim Highsmith, dem populärem Agile Guru, zu sagen '*Agilität ist die Fähigkeit, Veränderungen sowohl zu kreieren wie auf sie zu reagieren, um in einem wirtschaftlich turbulenten Umfeld profitabel zu sein. Agilität ist die Fähigkeit das Gleichgewicht zwischen Flexibilität und Stabilität zu halten*'.

Es ist wichtig festzuhalten, dass Agilität nicht nur Flexibilität ist, sondern die Ausgewogenheit von Flexibilität und Stabilität.

A.2.1 Der Aufstieg von Agile

Rasante Veränderungen in Technologie, Systemen und Verhaltensweisen haben traditionelle Managementmodelle ineffektiv werden lassen und so den Weg für Methodiken mit Agile Wertvorstellungen geebnet.

Einige Faktoren, die maßgeblich zum Aufstieg von Agile beigetragen haben, beinhalten:

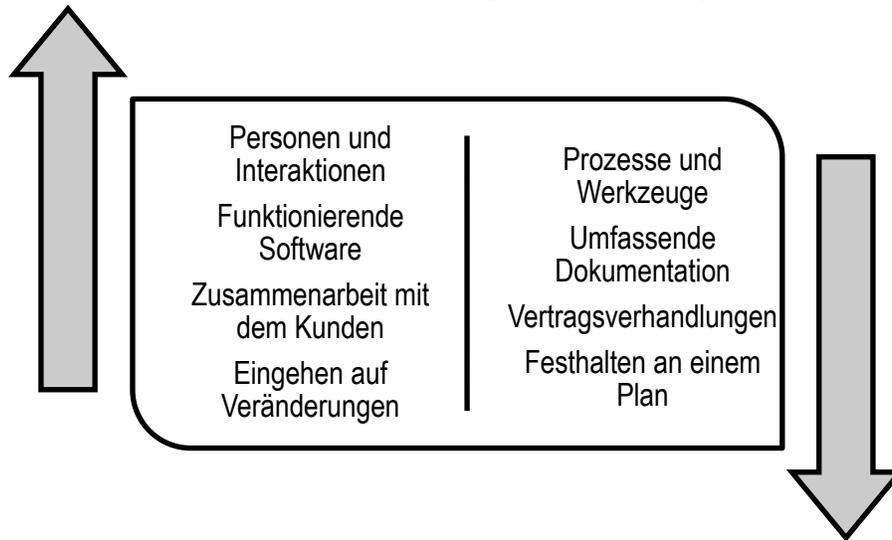
1. Wettbewerbsintensive Märkte
2. Niedrigere Kosten für Experimente
3. Ständiger Fortschritt der Technologie
4. Kürzere Produktlebenszyklen
5. Wissensarbeiter ersetzen Facharbeiter

Agile Techniken entwickelten sich mit der Zeit infolge der offensichtlichen Mängel in den traditionellen Wasserfall Techniken, welche darauf Wert legen, eine umfassende Vorausplanung durchzuführen und einmal erstellte Pläne wie geplant durchzuführen. Agile setzt auf adaptive Planung sowie iterative Lieferung und konzentriert sich auf Personen sowie funktionierende Software, um Aufgaben effektiv zu erledigen. Obwohl anpassungsfähige und inkrementelle Methoden seit den 1950er Jahren existieren, werden nur Methoden, die mit dem Agile Manifest übereinstimmen als Agile bezeichnet.

A.3 Das Agile Manifest

Im Jahr 2001 traf sich eine Gruppe von 17 Computer Gurus, Entwicklern und Managern, um über einfache Entwicklungsmethoden zu sprechen und veröffentlichte „Das Agile Manifest“. Es wurde als ein Hilfsmittel zur Festlegung einfacher Richtlinien für jede Agile Methodologie entwickelt. Das Manifest lautet wie folgt:

Wir zeigen bessere Wege auf, Software zu entwickeln,
indem wir es selbst tun und anderen dabei helfen, es zu tun.
Durch unsere Arbeit sind wir zu folgender Erkenntnis gekommen:



Das heißt, obwohl die Punkte auf der rechten Seite durchaus wichtig sind, halten wir die Punkte auf der linken Seite für wichtiger.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

© 2001, die oben erwähnten Autoren

Mit dieser Anmerkung darf diese Deklaration frei und in jeglicher Form kopiert werden, aber ausschließlich in seiner kompletter Form).

Die Erlaubnis zur Reproduktion wurde durch die oben genannten Autoren mittels Bekanntmachung unter <http://agilemanifesto.org> gegeben.

Die vier durch das Agile Manifest hervorgehobenen Gegenüberstellungen werden nachstehend erklärt:

1. **Personen und Interaktionen** sind wichtiger als **Prozesse und Werkzeuge**.
Auch wenn Prozesse und Werkzeuge zu einem erfolgreichen Abschluss eines Projekts beitragen, liegt es letztlich an den Menschen, die ein Projekt durchführen, daran teilhaben und es umsetzen, wie erfolgreich diese Prozesse und Werkzeuge angewandt werden. Die Schlüsselfiguren eines jeden Projekts sind die Menschen und der Schwerpunkt sollte daher auf ihnen und ihren Interaktionen liegen.
2. **Funktionierende Software** ist wichtiger als **umfassende Dokumentation**.
Auch wenn Dokumentation für jedes Projekt notwendig und hilfreich ist, konzentrieren sich viele Teams auf das Sammeln und Aufzeichnen qualitativer und quantitativer Beschreibungen von zu liefernden Ergebnissen, wobei der echte Wert für den Kunden primär eine funktionierende Software ist. Dementsprechend fokussiert Agile den ganzen Produktlebenszyklus hindurch auf die inkrementelle Lieferung funktionierender Software.
3. **Zusammenarbeit mit dem Kunden** ist wichtiger als **Vertragsverhandlungen**.
Traditionell werden Kunden als außenstehende Spieler angesehen, die hauptsächlich am Anfang und Ende des Produktlebenszyklus involviert sind und deren Verhältnis auf Verträgen und deren Erfüllung basiert. Agile glaubt an gemeinsame Werte, bei dem der Kunde als Partner angesehen wird. Das Team und der Kunde arbeiten zusammen, um das Produkt zu entwickeln und auszubauen.
4. **Eingehen auf Veränderungen** ist wichtiger als **Festhalten an einem Plan**.
Im heutigen Markt, wo Kundenanforderungen, verfügbare Technologien und Geschäftsmuster im ständigen Wandel sind, ist es unerlässlich einen anpassungsfähigen Ansatz zur Produktentwicklung zu haben, der den Einbau von Änderungen sowie einen kurzen Produktlebenszyklus ermöglicht, anstatt den Schwerpunkt auf der Verfolgung eines Plans zu legen, der mit potenziell veralteten Daten erstellt wurde.

Die mit dem Agile Manifest verbundenen Prinzipien werden nachstehend aufgeführt.

A.3.1 Prinzipien des Agile Manifests

Die 12 Prinzipien der Agile Manifesto von Fowler und Highsmith (2001) sind:

1. Unsere höchste Priorität ist es, den Kunden durch frühe und fortgesetzte Lieferung wertvoller Software zufrieden zu stellen.
2. Geänderte Anforderungen, selbst spät in der Entwicklung, sind erwünscht. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Regelmäßige Lieferung funktionierender Software--alle zwei Wochen bis hin zu alle zwei Monate -- eine kurze Zeitspanne wird bevorzugt.
4. Geschäftsleute und Entwickler müssen während des gesamten Projekts täglich zusammen arbeiten.
5. Baue das Projekte um motivierte Individuen herum auf. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue Ihnen, dass sie die Arbeit erfolgreich erledigen.
6. Die wirksamste und effektivste Methode Informationen an und innerhalb des Entwicklungsteams zu vermitteln ist von Angesicht zu Angesicht.
7. Eine funktionierende Software ist das wichtigste Maß des Fortschritts.
8. Agile Prozesse fördern eine nachhaltige Entwicklung. Die Sponsoren, Entwickler und Anwender sollten ein gleichmäßiges Arbeitstempo auf unbestimmte Zeit beibehalten können.
9. Ständiges Augenmerk auf technische Exzellenz und ein gutes Design verstärkt Agilität.
10. Einfachheit--die Kunst, die Menge nicht getaner Arbeit zu maximieren --ist essenziell.
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team darüber wie es effektiver werden kann und passt sein Verhalten dementsprechend an.

A.3.2 Interdependenzerklärung

Die Agile Interdependenzerklärung zum Projekt Management wurde Anfang 2005 von einer Gruppe von 15 Agile Projekt Leadern als Ergänzung zum Agile Manifest geschrieben. Es zählt sechs Managementwerte auf, die notwendig sind, um ein Agile Entwicklungsmentalität zu bestärken.

Die Erklärung hebt hervor, dass Projektteams, Kunden und andere Business Stakeholdern voneinander abhängig und miteinander verbunden sind and das anerkennen müssen, um erfolgreich zu sein. Die Werte als solche sind auch voneinander abhängig.

"Wir ...

erhöhen die Anlagenrendite indem wir unser Augenmerk auf einen kontinuierlichen Wertfluss legen.

liefern verlässliche Ergebnisse durch regelmäßige Interaktionen mit dem Kunden und gemeinsame Übernahme von Verantwortung.

erwarten Ungewissheit und behandeln diese mit Iterationen, Voraussicht und Adaptation.

lassen Kreativität und Innovation freien Lauf indem wir anerkennen, dass Menschen die ultimative Quelle für Wertschöpfung sind und wir ein Umfeld erschaffen in dem sie etwas bewegen können.

erhöhen die Leistungsfähigkeit durch Gruppenverantwortung für Ergebnisse und geteilte Verantwortung für Teameffektivität.

verbessern Effektivität und Zuverlässigkeit durch situationsspezifische Strategien, Prozesse und Praktiken."

Anderson, D., Augustine, S., Avery, C., Cockburn, A., Cohn, M., et al. 2005

A.4 Agile Methoden

Eine Anzahl von Agile Methoden haben ihren Ursprung und gewannen an Schwung in den 1990er und den frühen 2000er Jahren. Während die verschiedenen Agile Methoden in verschiedenen Aspekten voneinander abweichen, stammen ihre Gemeinsamkeiten aus dem Festhalten am Agile Manifest.

Die folgenden Agile Methoden werden nachstehend kurz angesprochen:

1. Lean Kanban
2. Extreme Programmierung
3. Crystal Methoden
4. Dynamische Systementwicklungsmethode
5. Featuregetriebene Entwicklung (Feature Driven Development)
6. Testgetriebene Entwicklung (Test Driven Development)
7. Adaptive Software Entwicklung
8. Agile Unified Process
9. Domain Getriebene Entwicklung (Domain Driven Development)

A.4.1 Lean Kanban

Das Lean (schlanke) Konzept optimiert das System einer Organisation, wertvolle Ergebnisse basierend auf ihren Ressourcen, Bedürfnissen und Alternativen zu produzieren und gleichzeitig Abfall zu reduzieren. Abfall kann durch Produktion von nicht benötigten Dingen, mangelnde Lernfähigkeit oder Praktiken, die den Prozess behindern, entstehen. Da diese Faktoren dynamischer Natur sind, bewertet eine schlanke Organisation das gesamte System und verfeinert die Prozesse kontinuierlich. Die Grundlage von Lean besagt, dass die Verkürzung der Zeitspanne eines jeden Zyklus (Iteration) zu einer erhöhten Produktivität führt, da Verzögerungen vermindert werden, Fehler schon früh entdeckt werden und folglich den Gesamtaufwand zur Erledigung der Aufgabe reduziert wird. Lean Prinzipien wurden erfolgreich in der Software Entwicklung angewandt.

Kanban (wörtlich übersetzt Aushängeschild) tritt für den Gebrauch von visuellen Hilfsmitteln ein, um die Produktion zu unterstützen und zu verfolgen. Das Konzept wurde von Toyota Motor Corporation, im Toyota Produktionssystem (TPS), eingeführt. Die Benutzung visueller Hilfsmittel ist sehr effektiv und gehört mittlerweile zur gängigen Praxis. Beispiele dafür sind Aufgabenkarten, Scrum Tafeln und Burndown Charts. Beide Methoden erregten durch ihren Einsatz bei Toyota, einem Vorreiter des Prozessmanagements, Aufmerksamkeit. Lean Kanban integriert die Benutzung der Visualisierungsmethoden, wie von Kanban beschrieben, mit den Prinzipien von Lean, um ein visuelles, inkrementelles und evolutionäres Prozessmanagementsystem zu erstellen.

A.4.2 Extreme Programmierung

Extreme Programmierung (XP) hat seinen Ursprung in der Chrysler Corporation und gewann in den 1990er Jahren an Zugkraft. XP ermöglicht es zu verhindern, dass die Kosten einer Softwareänderung nach und nach drastisch ansteigen. Die Hauptelemente von XP schließen inkrementelle Entwicklung, flexibles Planen, automatische Testcodes, verbale Kommunikation, ein sich ständig weiterentwickelndes Design, enge Zusammenarbeit und das enge Einbeziehen dessen, was alle Beteiligten langfristig und kurzfristig anspricht, mit ein.

XP legt Wert auf Kommunikation, Rückmeldungen, Einfachheit und Mut. Die verschiedenen Rollen im XP Ansatz schließen den Kunden, den Entwickler, den Nachfolger und den Trainer ein. XP schreibt verschiedene Programmierungs-, Entwickler- und Geschäftspraktiken sowie Veranstaltungen und Artefakte vor, um ein effektives und effizientes Entwickeln zu erreichen. XP wurde aufgrund seiner gut definierten Entwicklungsverfahren weit verbreitet angenommen.

A.4.3 Crystal Methode

Die Crystal Methodik der Software Entwicklung wurde von Alistair Cockburn in den frühen 1990er Jahren eingeführt. Crystal Methoden sind dazu angedacht, Menschen in den Mittelpunkt zu stellen, unkompliziert und leicht anpassbar zu sein. Da Menschen im Vordergrund stehen, sind die Entwicklungsprozesse und Werkzeuge nicht fest, sondern eher den Anforderungen und Charakteristiken des Projekts angepasst. Das Farbspektrum wird verwendet, um die Crystal Variante für das Projekt festzulegen. Faktoren wie Komfort, verfügbares Geld, erforderliches Geld und Leben spielen eine wichtige Rolle, um das „Gewicht“ der Methodologie zu bestimmen, welche in verschiedenen Farben des Spektrums dargestellt wird. Die Crystal Familie ist unterteilt in Crystal Clear (Klar/Transparent), Crystal Yellow (Gelb), Crystal Orange, Crystal Orange Web, Crystal Red (Rot), Crystal Maroon (Kastanienbraun), Crystal Diamond (Diamant) und Crystal Sapphire (Saphirblau).

Alle Crystal Methoden haben vier Rollen – der hochrangige Sponsor, der leitende Designer, Entwickler und erfahrene Anwender. Crystal Methoden empfehlen verschiedene Strategien und Techniken um Agilität zu erreichen. Ein Crystal Projektzyklus besteht aus dem Chartering, Lieferzyklus und Abschluss.

A.4.4 Dynamische Systementwicklungsmethode

Das Rahmenwerk für die Dynamische Systementwicklungsmethode (DSDM)- wurde anfänglich in 1995 veröffentlicht und wird vom DSDM Konsortium verwaltet. DSDM setzt Qualität und Aufwand, in Bezug auf Kosten und Zeit, zu Beginn fest und passt die zu liefernden Ergebnisse des Projekts den gesetzten Kriterien an, indem die zu liefernden Ergebnisse in musts (müssen), shoulds (sollten), coulds (könnten) und won't have (werden wir nicht haben) (MoSCoW), eingeteilt werden.

DSDM ist eine System-orientierte Methode mit sechs unterschiedlichen Phasen – die Vor-Projekt-Phase, die Machbarkeitsphase, die Grundlagenphase, die Erkundungs- und Entwicklungsphase, die Einsatzphase und die Nutzeneinschätzungsphase.

A.4.5 Featuregetriebene Entwicklung (Feature Driven Development)

FDD wurde 1997 von Jeff De Luca erfunden und arbeitet nach dem Prinzip ein Projekt in kleine, Kundenmehrwert liefernde Funktionen aufzuteilen, die in weniger als zwei Wochen geliefert werden können. FDD hat zwei Grundprinzipien - die Softwareentwicklung ist eine menschliche Tätigkeit und eine Kundenmehrwert Funktionalität.

FDD definiert sechs Hauptrollen - den Projektmanager, Chefarchitekt, Entwicklungsmanager, Chefprogrammierer, Class Owners und die Domänenexperten, zuzüglich einer Anzahl von unterstützenden Rollen. Der FDD Prozess ist iterativ und besteht aus der Entwicklung eines Gesamtmodells und der Erstellung einer Feature-Liste bevor dann Feature für Feature geplant, entworfen und entwickelt wird.

A.4.6 Testgetriebene Entwicklung (Test Driven Development)

Die testgetriebene Entwicklung (TDD), auch bekannt als Test zuerst Entwicklung, wurde von Kent Beck, einem der Entwickler von XP, eingeführt. Sie ist eine Softwareentwicklungsmethode die es beinhaltet, zuerst automatische Testcodes zu schreiben und dann den minimalen Code zu entwickeln, der diesen Test zu einem späteren Zeitpunkt bestehen wird. Das komplette Projekt wird in kleine, am Kundenwert orientierte Features aufgeteilt, die in einem so kurz wie möglich gehaltenen Entwicklungszyklus erstellt werden müssen. Basierend auf den Anforderungen und Spezifikationen des Kunden werden Tests geschrieben. Die Tests, die in der oben genannten Phase erstellt werden, werden dafür angewandt, um den Produktionscode zu entwerfen und zu schreiben.

TDD kann in die folgenden zwei Ansätze eingestuft werden: ATDD – akzeptanztestgetriebene Entwicklungen erfordern einen ausgeprägten Akzeptanztest und DTDD – die Entwickler TDD beinhaltet einen einzelnen Entwicklertest zu schreiben. TDD ist wegen ihrer vielen Vorteile, z.B. schnelle und verlässliche Ergebnisse, kontinuierliche Rückmeldungen und eine verkürzte Fehlerbeseitigungszeit, sehr populär geworden.

A.4.7 Adaptive Software Entwicklung (Adaptive Software Development)

Adaptive Software Development geht auf die Rapid Application Development Arbeit von Jim Highsmith und Sam Bayer zurück. Die Höhepunkte von ASD sind die kontinuierliche Anpassung von Prozessen für die auszuführende Arbeit, die Bereitstellung von Lösungen zu Problemen, welche in großen Projekten auftauchen und der iterativen, inkrementellen Entwicklung, mit kontinuierlich neu erstellten Programmversionen als Prototypen.

Als ein risikogetriebener und änderungstoleranter Entwicklungsansatz glaubt ASD, dass ein Plan keine Unsicherheiten und Risiken einräumen darf, da diese auf einen fehlerhaften und gescheiterten Plan hinweisen. ASD ist feature basiert und zielorientiert. Die erste Phase der Entwicklung in ASD ist das Spekulieren, im Gegensatz zur Planung wie in den anderen agilen Methodiken, gefolgt von den Phasen des Zusammenarbeitens und Lernens.

A.4.8 Agile Unified Process

AUP ist eine vereinfachte Version von IBM's Rational Unified Process und wurde von Scott Ambler entwickelt. AUP kombiniert branchenbewährte Agile Techniken, wie testgetriebene Entwicklung, Agile modeling und eine Datenbank-Redesign, um ein funktionierendes Produkt bester Qualität zu liefern.

AUP formt seine Prozesse und Techniken an den Werten von Einfachheit, Agilität, Anpassbarkeit, Selbstorganisation, Unabhängigkeit von Werkzeugen und legt den Schwerpunkt auf Tätigkeiten von hohem Wert. Die AUP Prinzipien und Werte werden in Phasen in die Tat umgesetzt – Einführung, Ausarbeitung, Erzeugung und Übergang.

A.4.9 Domänen Getriebener Entwurf (Domain-Driven Design)

Domain-Driven Design ist ein Agile Entwicklungsansatz, gedacht für die Handhabung komplexer Entwürfe, geknüpft an ein sich weiterentwickelndes Modell. Es wurde von Eric Evans im Jahre 2004 vorgeschlagen und dreht sich um den Entwurf einer Kerndomäne. Die Domäne kann als Bereich von Aktivitäten definiert werden, auf den der Anwender ein Programm oder eine Funktionalität anwendet. Viele solcher Bereiche werden gebündelt und ein Modell wird erstellt, welches ein Abstraktionssystem ist, das benutzt werden kann, um das Gesamtprojekt zu entwerfen und die Probleme im Zusammenhang mit den gebündelten Domänen zu lösen. Die Kernwerte von DDD beinhalten Domänen-orientiertes, modellgetriebenes Entwerfen, eine ubiquitäre Sprache und einen begrenzten Kontext.

In der Design orientierten Entwicklung wird eine ubiquitäre Sprache eingeführt und die Domäne wird geformt. Danach folgt die Gestaltung, Entwicklung und Prüfung. Das Modell der Domäne wird bis zur Zufriedenstellung verfeinert und überarbeitet.

ANHANG B. AUTOREN UND REZENSENTEN DES SBOK®

Dieser Anhang listet, innerhalb ihrer Gruppen in alphabetischer Reihenfolge, die Namen der Personen auf, die zur Entwicklung und Produktion des SBOK® beigetragen haben.

SCRUMstudy™ ist allen diesen Personen für ihre kontinuierliche Unterstützung dankbar und erkennt ihren Beitrag zu der Entwicklung dieser Wissenssammlung an.

B.1 HAUPTAUTOR

Tridibesh Satpathy

B.2 Co-Autoren und Fachexperten

Winfried Hackmann

Gaynell Malone

Ruth Kim

Buddy Peacock

Deepak Ramaswamy

Quincy D. Jordan

J. Drew Nations

Karen Lyncook

Jaimie M. Rush

Elizabeth Lynne Warren

Gaurav Garg

Dipaka Patra

Ahmed Touseefullah Siddiqui

Nikhil Kumar

B.3 Mitwirkende und Rezensenten Team

Abdelnaser Dwaikat, Btech, Mtech

Abhijit Daayma, MBA, SMC

Aimee Norman, SMC

Alec Vasquez, SMC
Alejandro Córdova, MBA, PMP, PMI-RMP, SCT
Angela Mascarenas, BSC, SMC, SPOC
Aniruddha Banerjee, SMC, SFC
Antonio Marcias, SPOC, SFC
Anu Ravi, BE, SPS, PMP, ITIL, PSM1
Arturo Velazquez, MSC, SMC, SFC
Barbara Siefken, MBA, PMP, ITIL, CSM
Bobbie Green, MBA
Brian Rubin, SPOC
Bryan Lee Perez, MS, PMP, CSM, SSGB, SMC
Carlos Acuña, MBA, PMP, PgMP, CBAP, RMP, SCT
Charles J. Quansah, MSIT, CCNA, PMP, ITIL
Charles Letterman, SMC
Chrys Thorsen, CCSI, MCSE, CISSP
Corey Bailey, PMP, ITIL, SMC, SAMC
Corky Henderson, MBA, PMP, SMC, SSGB
Cristian Mauricio Avila Patarroyo, SPOC
Damien Lee, PMP, ACP, SMC, SPOC
David Soden, SMC
Deepa K SMC, SPOC, SFC
Derik Stalls, PMP, ITIL, SPOC
Dusan Kamenov, PMP, PRINCE2, CSM, SFC
Efetobore Adebayo Omadevuae, MBA, ITIL, SAFe-SA, SMC
Enrique Vanegas, SMC, SPOC
Frances Mary Jo Tessler, PMP
Frank Quinteros, MBA, PMP, PMI-ACP
Gabriel Joseph, MSC, SMC, SAMC, SPOC
Ganesh Watve, MBA, PMP, SCT
George Hanosh, BSC, SMC, SPOC
Gerald Varghese, SMC, SPOC
Girish Kulkarni, MSc, ITIL, PRINCE2, CSM
Ian Glenister, BA, PRINCE2, PMP, SMC, SPOC
Ignacio Navarro Zaragoza, PMP, ITIL, LEAN IT, SCT
Inder Mohan Singh, SMC, SFC
Isaiah Rajan, PMP, MBA, SMC, SFC
James Cajuste, SMC, SFC
James McHarry, SMC
James Pruitt, ITIL, SMC, SPOC
James Rafferty, SPOC
Jared Smuli, SMC

Javier González, MBA, PMI-ACP, ITIL, MCP, SCT
Jeff Blitstein, MBA, PMP, ACP, SPOC
Jim Huckin, PMP, SMC
Jo Pereira, SMC
Joe Schofield, SAFe- SA, LSSBB, CSQA, CMMI, SCT
Jose Antonio Pineda Mora, MBA, PMP, ITIL, SCT
Jose Nunez, PhD, PE, PMP
Joshua Adelankun, SMC, SPOC
Juan Alberto Marques Rodriguez, SPOC
Juan Carlos Linares, PMP, SMC
Katherine Ricker, PMP, SFC, SPOC
Kuljeet Singh Sarna, MBA, PRINCE2, PMI-ACP, CISSP, SCT
Lachlan McGurk, PMP, ITIL, SSBB, SMC
Lennon Burhannan, SMC, SAMC, SPOC, SFC, SSMC
Lucy Vorpahl, BBA, SMC, SPOC, SFC
Madhuresh Kumar Mishra, PMP, PRINCE2, ITIL, SMC, SPOC
Magaline D. Harvey, MBA, PMP, SMC
Mariela Laborde, MBA, SMC, SPOC, SFC
Meena Elangovan, BE, PMP, SSGB, PSM
Melissa Lauro, MA, SMC, SAMC
Melvin Wofford Jr., BSc, PMP
Michael Harmon MBA, ITIL, SMC, SFC
Michael Rauch, MBA, PMP, ACP, PSM1
Michael Sanchez, SPOC
Michael W Madigan, SMC, SAMC, SPOC, SFC
Michelle Wilkinson, MBA, PMP, SMC
Mike Tomaszewski, PgMP, PMP, MBA
Mimi LaRaque, PMP, PMI-ACP
Miriam Kirkaldie, SMC
Mitch Malloy, SPC, RTE, SMC, SPOC
Monica Strazzante, PMP, SMC
Morris Feigel, PMP, PRINCE2, SPC, ITIL, PSPO1
Muminul Haque, MSC, SMC, SFC
Nadra Rafee, PMP, SAFe-SA, SMC
Neha Mishra, BBA, MBA,
Nichole Thompson, MBA, SPC, SMC
Nikhil Bhargava, BE, MBA, SMC, SPOC
Obi Nwaojigba, MBA, PMP, ITIL, SPOC
Olatunde Badmus, MBA, PMP, SMC
Olumide Idowu, PMP, ACPC, SMC
Oscar Esquivel, BSC, SMC, SFC

Paul de Cunzo, SMC, SPOC
Prof. Dr. Akram Hassan, MBA, PMP, RMP, SFC
Ranjit Majumda, SMC, SPOC, SFC
Raul Caban, ITIL, SMC, SPOC
Ravi Kumar Kalose, MCA, PGDBA, PMP, SCT
Ravneet Kaur, MBA, SMC, RTO
Richard Mather, MSc, PG
Rima Vyas, SMC
Robert Lamb, PMP, MCCT, MCSE
Romil Desai MBA, PMP, LSSBB, SMC, SFC
Ron Villmow, MSc, MCT
Sandra A. Strech, PMP
Sandy (Sanjukta) Banerjee, SMC
Santosh Heroorker, MS, SMC, SFC
Saurabh Gupta, BE, SAFe, AINS
Sean McVeigh, SMC
Seun Odunlami, LPM, SAFe-SA, SMC, SFC
Sheri Palmer, PgMP, PMP, SMC, SFC
Sheryl Cattrell, SPOC, SFC
Simon Robertson, PMP, MSP, SMC, SAMC, SCT
Sohini Banerjee, BA, SMC
Sourabh Sharma, SMC
Srikanth PV, MBA, CMA, PMP, SSGB, SCT
Srinivas Reddy Kandi, MBA, MCA, IOT
Steve Versteer, SMC
Sudheer Vankadara, MBA, SDC, SMC, SAMC, SPOC
Sunil Krishnan, MBA, SMC
Syed Ashraf, BSc, MCA, PRINCE2
Thomas Nelson Woltz, SMC, SFC
Tommie L. Sherrill, MBA, PMP, SMC
Tracey Branch, SMC
Tushar Purohit, PRINCE2, ITIL, SSGB, SMC
Vicente Manuel Guerra Hernández, SDC, SMC
Vince Belanger, BAsC, PMP
Vinod Kumar, MCA
Yogaraj Mudalgi, BA, SMC
Yvonne Van Horn, SPOC

REFERENZEN

Anderson, D., Augustine, S., Avery, C., Cockburn, A., Cohn, M., DeCarlo, D., Fitzgerald, D., Highsmith, J., Jepsen, O., Lindstrom, L., Little, T., McDonald, K., Pixton, P., Smith, P., e Wysocki, R. (2005) "Declaration of Interdependence," acessada em Setembro de 2013, <http://www.pmdoi.org/>.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001) "Manifesto for Agile Software Development," acessada em Setembro de 2013, <http://agilemanifesto.org/>.

Fellers, G. (1994) *Why Things Go Wrong: Deming Philosophy In A Dozen Ten-Minute Sessions*. Gretna, LA: Pelican Publishing.

Greenleaf, R. K. (1977) *Servant Leadership: A Journey into the Nature of Legitimate Power and Greatness*. Mahwah, NJ: Paulist Press.

Kano, N., Seraku, N., Takahashi, F., and Tsuji, S. (1984) "Attractive Quality and Must Be Quality." *Quality*, 14 (2): 39–48.

Leffingwell, D. and Widrig, D. (2003) *Managing Software Requirements: A Use Case Approach, 2nd ed.* Boston: Addison-Wesley.

Maslow, A. H. (1943) "A Theory of Human Motivation." *Psychological Review*, 50 (4): 370–396.

McGregor, D. (1960) *The Human Side of Enterprise*. New York: McGraw-Hill.

Patton, J. (2005) "It's All in How You Slice." *Better Software*, January: 16–40.

Takeuchi, H. and Nonaka, I. (1986) "The New New Product Development Game." *Harvard Business Review*, January–February: 137–146.

Glossar

100-Punkte-Methode

Die 100-Punkte-Methode wurde von Dean Leffingwell und Don Widrig (2003) entwickelt. Sie sieht vor, dem Kunden 100 Punkte zur Verfügung zu stellen, mit denen dieser die für ihn wichtigsten Features bewerten kann.

Akzeptierte zu liefernde Ergebnisse

Zu liefernde Ergebnisse, die die User Story Abnahmekriterien erfüllen, werden vom Product Owner akzeptiert. Sie gelten als akzeptierte zu liefernde Ergebnisse, die bei Bedarf an den Kunden freigegeben werden können.

Adaptierung

Adaptierung findet statt, indem das Scrum Kernteam und der bzw. die Business Stakeholdern durch Transparenz und Inspektion lernen und dann entsprechende Verbesserungen ihrer Arbeiten einführen.

Vergleichende Abschätzung

Vergleichende Abschätzung (Affinity Estimation) ist eine Technik zur schnellen Abschätzung einer großen Anzahl von User Storys. Die User Storys werden dazu als klein, mittel oder gross kategorisiert. Die Abschätzung erfolgt in Form von Story Points, die eine relative Größe anzeigen. Einige der wichtigsten Vorteile dieser Methode bestehen darin, dass der Prozess sehr transparent, jedem ersichtlich und einfach durchzuführen ist.

Vereinbarte Verbesserungsmaßnahmen

Vereinbarte Verbesserungsmaßnahmen gelten als primärer Output beim Sprint Retrospektions-Meeting. Sie definieren die Verbesserungsmaßnahmen, die sich das Team zur Problembehandlung und Prozessverbesserung hat einfallen lassen, um so die Leistung in zukünftigen Sprints zu verbessern.

Genehmigung, Abschätzung und Zusage von User Storys

In diesem Verfahren genehmigt der Product Owner User Storys für einen Sprint. Anschließend schätzen der Scrum Master und das Scrum Team den notwendigen Aufwand ab, um die in der User Story beschriebene Funktionalität zu entwickeln. Am Ende verpflichtet sich das Scrum Team dazu, die Kunden-Anforderungen in Form von genehmigten, abgeschätzten und zugesagten User Storys zu erfüllen.

Genehmigte Change Requests

Genehmigte Change Requests sind neue oder geänderte Anforderungen, deren Aufnahme ins priorisierte Backlog freigegeben worden ist. Gelegentlich können Genehmigte Change Requests von Programm- oder Portfoliomanagern veranlasst worden sein. Alle genehmigten Change Requests sind Kandidaten für eine Implementierung in zukünftigen Sprints.

Genehmigte, abgeschätzte und zugesagte User Storys

Für die User Storys, die als Input für diesen Prozess dienen, gibt es bereits high-level Abschätzungen von den Prozessen "Priorisiertes Produkt Backlog Erstellen" und "User Storys Erstellen".

Anhand dieser high-level Abschätzungen genehmigt der Product Owner User Storys für den Sprint. Einmal genehmigt werden User Storys vom Team anhand verschiedener Abschätztechniken detaillierter abgeschätzt. Nach der Abschätzung sagt das Team eine Untermenge an genehmigten und abgeschätzten User Storys zu in der Annahme, diese im nächsten Sprint vollständig implementieren zu können. Diese User Storys gelten als genehmigte, abgeschätzte und zugesagte User Storys und werden Teil des Sprint Backlogs."

Selbstbewusster Leader

Selbstbewusste Leader stellen sich Problemen und treten selbstsicher auf, um sich mit Respekt Autorität zu verschaffen.

Zugeordnete Aktionspunkte und Fertigstellungstermine

Sobald die vereinbarten Verbesserungsmaßnahmen ausgearbeitet und verfeinert worden sind, können Aktionspunkte zur Implementierung der Verbesserungen vom Scrum Team in Erwägung gezogen werden. Jeder Aktionspunkt erhält einen festen Fertigstellungstermin.

Autokratischer Leader

Autokratische Leader treffen Entscheidungen auf eigene Faust und erlauben somit Teammitgliedern wenig bis keine Teilnahme an oder Auseinandersetzung mit der Entscheidungsfindung. Dieser Führungsstil sollte nur selten verwendet werden..

Automatisierte Software Tools

Bei automatisierten Software Tools handelt es sich um Software Tools, die zur Terminierung, Datensammlung und Distribution verwendet

Bessere Team Koordinierung

Das Scrum der Scrums Meeting fördert die Arbeitskoordinierung über mehrere Scrum Teams. Besonders wichtig ist dies im Fall von Aufgaben, bei denen Abhängigkeiten zwischen Teams bestehen.

Inkompatibilitäten und Diskrepanzen zwischen der Arbeit und den zu liefernden Ergebnissen unterschiedlicher Teams kommen schnell zum Vorschein. Dieses Forum ermöglicht Teams zudem, ihre Ergebnisse darzustellen und anderen Teams feedback zu geben.

Brainstorming / Brainstorming

Sitzungen, in denen relevante Business Stakeholdern und Mitglieder des Scrum Kern Teams anhand von Diskussionen und Erfahrungsaustausch offen Ideen austauschen. Diese Sitzungen werden in der Regel von einem Moderator geleitet..

Geschäftliche Rechtfertigung

Geschäftliche Rechtfertigung zeigt die Gründe für die Durchführung eines Projekts auf. Sie beantwortet die Frage 'Warum wird dieses Projekt benötigt?' Geschäftliche Rechtfertigung liefert den Antrieb für die gesamte Entscheidungsfindung in Bezug auf das Projekt.

Geschäftsbedarf

Geschäftsbedarf sind die vom Projekt zu erbringenden Geschäftsergebnisse, wie sie in der Projektvision dokumentiert sind.

Geschäftliche Anforderungen

Geschäftliche Anforderungen definieren, was geliefert werden muss, um Geschäftsbedarf zu erfüllen und Mehrwert für Business Stakeholderns zu liefern. Die Gesamtheit aller Erkenntnisse, die über verschiedene Werkzeuge wie Anwender oder Kundeninterviews, Fragebögen, JAD-Sitzungen, Gap-Analyse, SWOT-Analyse, und andere Meetings gewonnen wurden, hilft dabei eine bessere Perspektive der geschäftlichen Anforderungen zu bekommen und hilft auch bei der Erstellung des Priorisierten Produkt Backlogs."

Change Request(s)

Anfragen für Änderungen werden normalerweise als Change Requests übermittelt. Change Requests gelten nur dann als genehmigt, wenn sie formal genehmigt worden sind.

Chief Product Owner

Im Fall von großen Projekten ist es Aufgabe des Chief Product Owners, das gesamte priorisierte Produkt Backlog für das Projekt zu erstellen und zu pflegen. Er oder sie koordiniert Arbeiten zwischen den Product Ownern der Scrum Teams. Die Product Owner wiederum verwalten die ihnen zugehörigen Teile des priorisierten Produkt Backlogs.

Chief Scrum Master

Im Fall von großen Projekten ist es Aufgabe des Chief Scrum Masters, Scrum der Scrums (SoS) Meetings zu moderieren und Impedimente, die mehrere Teams beeinträchtigen, zu beseitigen.

Unterstützender Leader

Coachende und unterstützende Leader erteilen Anweisungen und unterstützen und beaufsichtigen anschließend ihre Teammitglieder, indem sie zuhören, Hilfestellung leisten, ermuntern und in Zeiten der Unsicherheit eine positive Einstellung aufzeigen.

Kollaboration

Kollaboration in Scrum bezieht sich darauf, wie das Scrum Kern Team untereinander und mit den Business Stakeholdern zusammenarbeitet, um die zu liefernden Ergebnisse des Projektes zu erstellen und zu validieren, so dass die Ziele, die in der Projekt Vision umschrieben sind, erreicht werden. Kollaboration findet dann statt, wenn ein Team zusammenarbeitet, um die Beiträge einzelner Teammitglieder zu einem Grösseren Ganzen zusammenzufügen.

Kollaborationsplan

Kollaboration ist ein äußerst wichtiges Element in Scrum. Im Kollaborationsplan wird erläutert, wie sich die verschiedenen Entscheidungsträger, Business Stakeholdern und Team Mitglieder engagieren und untereinander zusammenarbeiten.

Colocation

Unter Colocation versteht man die Zusammenlegung aller Scrum Kernteammitglieder an einen gemeinsamen Arbeitsort, um die daraus entstehenden Vorteile bezüglich Koordination, Problemlösung, Erfahrungsaustausch und Lernprozessen zu nutzen.

Kommunikationsplan

Dieser Plan spezifiziert die anzulegenden und zu pflegenden Datensätze während des Projekts. Dazu dient eine Vielzahl an Methoden der Vermittlung wichtiger Projektinformationen an die Business Stakeholdern. Der Kommunikationsplan definiert sowohl diese Methoden als auch die für diverse Kommunikationsaktivitäten Verantwortlichen.

Unternehmensmission

Mit der Unternehmensmission erstellt ein Unternehmen oder eine Organisation einen Rahmen, der wiederum als Leitfaden für die gesamte Entscheidungsfindung dient.

Unternehmensvision

Das Verstehen der Unternehmensvision trägt dazu bei, die Ziele der Organisation und das Zukunftspotential des Unternehmens im Fokus des Projekts zu halten. Der Product Owner kann die Unternehmensvision als Richtwert und Wegweiser bei der Erstellung des Project Vision Statements nutzen.

Täglichen Standup durchführen

Täglichen Standup Durchführen ist ein Prozess, in dem jeden Tag ein hochkonzentriertes Meeting mit einem harten zeitlichen Anschlag durchgeführt wird. Man spricht von diesem Meeting als Tägliches Standup Meeting. Es ist ein Forum für das Scrum Team, bei dem sich die Team Mitglieder gegenseitig über Ihren Fortschritt und jegliche Hindernisse informieren, denen sie gfls. gegenüberstehen.

Release-Planung Durchführen

In diesem Prozess überprüft das Scrum-Kernteam die im Priorisierten Produkt Backlog enthaltenen high-Level User Storys, um einen Release-Plan zu entwickeln. Dieser stellt im Wesentlichen einen schrittweisen Bereitstellungszeitplan dar, der den Business Stakeholdern zur Verfügung gestellt werden kann. In diesem Prozess wird auch die Länge der Sprints festgelegt.

Konfliktmanagement

Konflikt Management Techniken werden von Team Mitgliedern zur Lösung von Konflikten verwendet, die während eines Scrum Projekts entstehen. Konfliktquellen sind oftmals Zeitpläne, Prioritäten, Ressourcen, Hierarchie, technische Probleme, Vorgehensweisen, Persönlichkeiten und Kosten.

Kontinuierliche Verbesserung

Kontinuierliche Verbesserung ist ein Scrum Ansatz, bei dem das Team aus eigener Erfahrung und dem Engagement der Business Stakeholdern lernt, um das Priorisierte Produkt Backlog kontinuierlich bzgl. aller Änderungen der Anforderungen auf dem aktuellen Stand zu halten.

Kontinuierliche Wertrechtfertigung

Kontinuierliche Wertrechtfertigung bezieht sich auf die regelmäßige Bewertung des Geschäftlichen Werts um zu prüfen, ob die Fortführung des Projekts weiterhin gerechtfertigt und machbar ist.

Zum Scrum der Scrums zusammenkommen

In vorbestimmten Intervallen oder nach Bedarf kommen in diesem Prozess die Scrum Master oder andere Vertreter der verschiedenen Scrum teams zum "Scrum der Scrums" (oder Scrum of Scrums) Meeting zusammen und verfolgen ihre jeweiligen Fortschritte und Hindernisse sowie Abhängigkeiten zwischen Teams.

Kernrolle (n)

Kernrollen sind die Rollen, die zwingend zur Herstellung des mit dem Projekt verbunden Produkts erforderlich sind. Sie sind dem Projekt wirklich verpflichtet und außerdem für den Erfolg jedes Sprints innerhalb des Projekts sowie für den Erfolg des Projekts als Ganzes verantwortlich.

Zu liefernde Ergebnisse erzeugen

"Zu liefernde Ergebnisse erzeugen" ist der Prozess, bei dem das Scrum-Team an den Aufgaben im Sprint Backlog arbeitet um die zu liefernden Ergebnisse des Sprints zu erzeugen.

Priorisiertes Produkt Backlog erstellen

In diesem Prozess werden Epic (en) definiert und ausgearbeitet, dann priorisiert, um ein Priorisiertes Produkt Backlog für das Projekt zu erstellen. Die Done-Kriterien werden ebenfalls zu diesem Zeitpunkt festgelegt.

Projekt Vision erstellen

In diesem Verfahren wird die wirtschaftliche Begründung für das Projekt herangezogen, um eine Projektvision zu erstellen, die als Inspiration und Richtschnur für das gesamte Projekt dient. Der Product Owner wird in diesem Prozess festgelegt.

Sprint Backlog aktualisieren

In diesem Prozess hält das Scrum Kernteam Sprint Planungssitzungen ab, in denen die Gruppe ein Sprint Backlog erstellt, das alle Aufgaben enthält, die im Sprint abgeschlossen werden sollen.

Aufgaben erstellen

In diesem Prozess werden die Genehmigten, Geschätzten, und Zugesagten User Storys in spezifische Aufgaben aufgeteilt und in einer Aufgabenliste zusammengetragen. Oft wird zu diesem Zweck eine Aufgabenplanungssitzung gehalten.

User Storys Erstellen

In diesem Prozess werden User Storys und die damit verbundenen User Story Abnahmekriterien erstellt. User Storys werden in der Regel vom Product Owner geschrieben und sollen sicherstellen, dass die Anforderungen des Kunden deutlich und für die Business Stakeholdern verständlich dargestellt worden sind.

Kumulatives Flussdiagramm (CFD)

Ein kumulatives Flussdiagramm (CFD) ist ein nützliches Werkzeug für die Berichterstattung über und die Verfolgung der Projekt Performance. Es bietet eine einfache, visuelle Darstellung des Projektfortschritts zu einem bestimmten Zeitpunkt. Es wird normalerweise dazu verwendet, um einen High-Level Status des Gesamtprojekts bereitzustellen. Es ist nicht dazu da, tägliche Aktualisierungen für einzelne Sprints zu liefern.

Kunde

Beim Kunden handelt es sich um eine Einzelperson oder eine Organisation, die das/den aus dem Projekt entstandene Produkt, Service oder generelle Resultat erwirbt. Für jede Organisation kann es je nach Projekt sowohl interne Kunden (d.h. innerhalb derselben Organisation) als auch externe Kunden (d.h. außerhalb der Organisation) geben.

Kundenwertorientierte Priorisierung

Kundenwertorientierte Priorisierung legt größten Wert auf den Kunden und ist bestrebt, User Storys mit dem höchsten Wert zuerst zu implementieren. Die User Storys mit dem höchsten Wert werden identifiziert und an den Anfang des Priorisierten Produkt Backlogs bewegt.

Tägliches Standup Meeting

Unter dem Täglichen Standup Meeting versteht man ein kurzes tägliches Meeting mit einer Time-Box von 15 Minuten. Die Teammitglieder treffen zusammen, um von ihren Fortschritten berichten, indem sie die folgenden drei Fragen beantworten:

1. Was habe ich gestern fertiggestellt?
2. Was werde ich heute fertigstellen?
3. Vor welchen Impediments oder Hemmnissen (falls vorhanden) stehe ich zur Zeit?

Zerlegung

Zerlegung ist ein Werkzeug, wodurch high-level Aufgaben in low-level und mehr detaillierte Aufgaben aufgespalten werden. Die User Storys werden durch die Mitglieder des Scrum Teams in Aufgaben zerlegt. User Storys aus dem Priorisierten Produkt Backlog müssen soweit zerlegt werden, dass dem Scrum-Team angemessene Informationen zur Verfügung stehen, um anhand der in der Aufgabenliste erwähnten Aufgaben die zu liefernden Ergebnisse zu erstellen.

Delegierender Leader

Delegierende Leader sind in einen Großteil der Entscheidungsfindung eingebunden; jedoch geben sie einen Teil der Planungs- und Entscheidungsverantwortung an Teammitglieder weiter, insbesondere, wenn diese fähig sind, Aufgaben kompetent umzusetzen. Dieser Führungsstil eignet sich für Situationen, in denen der Leader mit speziellen Projektdetails konform geht und Zeit begrenzt ist.

Sprint Vorführen and Validieren

In diesem Prozess führt das Scrum Team die erstellten Ergebnisse des Sprints dem Product Owner und den relevanten Business Stakeholdern in einem Sprint Review Meeting vor.

Abhängigkeitsermittlung

Sobald das Scrum-Team User Storys für einen gegebenen Sprint ausgewählt hat, sollte es danach alle Abhängigkeiten in Betracht ziehen, sowohl die Verfügbarkeit von Menschen als auch technische Abhängigkeiten. Abhängigkeiten ordnungsgemäß zu dokumentieren hilft den Scrum-Teams dabei, die relative Reihenfolge zu bestimmen, in der Aufgaben ausgeführt werden sollten, um die zu liefernden Ergebnisse des Sprints zu erzeugen. Abhängigkeiten heben auch die Beziehung und Interaktionen hervor, die zwischen verschiedenen Aufgaben bestehen (sowohl innerhalb eines Scrum Teams, das an einem gegebenen Sprint arbeitet oder mit anderen Scrum Teams am Projekt).

Entwurfsmuster

Entwurfsmuster bieten eine formale Möglichkeit, Lösungen von Designproblemen in einem bestimmten Fachgebiet festzuhalten. Diese Muster halten sowohl den Prozess als auch die tatsächliche Lösung fest und können im späteren Verlauf erneut genutzt werden, um Entscheidungsfindung und Produktivität zu verbessern.

Epics entwickeln

In diesem Prozess dient die Projektvisionserklärung als Grundlage dazu, Epics zu entwickeln. Anwendergruppen Meetings können abgehalten werden, um Epics zu entwickeln.

Vertrag zur Entwicklung in Phasen

Nach erfolgreichem Abschluss eines Releases stellt dieser Vertrag monatlich oder quartalsweise Geldmittel zur Verfügung. Es gibt den Kunden und Lieferanten einen Anreiz und garantiert dass das finanzielle Risiko des Kunden auf diese bestimmte Zeitspanne begrenzt ist, da erfolgreiche Releases nicht finanziert werden.

Anweisender Leader

Anweisende Leader weisen ihre Teammitglieder an, welche Aufgaben zu leisten sind sowie auf welche Art und zu welchem Zeitpunkt diese auszuführen sind.

Nicht-zwingende Abhängigkeiten

Nicht-zwingende Abhängigkeiten sind Abhängigkeiten, die nach Wahl in den Arbeitsfluss platziert werden. Nicht-zwingende Abhängigkeiten werden in der Regel durch das Scrum-Team auf der Grundlage von Erfahrungen oder von besten Praktiken in einem bestimmten Feld oder einer bestimmten Domäne festgelegt.

Done-Kriterien

Done Kriterien sind eine Reihe von Regeln, die für alle User Storys gelten. Eine klare Definition von Done (erledigt) ist entscheidend, weil es die Mehrdeutigkeit aus den Anforderungen entfernt und dabei hilft, dass das Team sich an verbindliche Qualitätsnormen hält. Diese klare Definition wird dazu verwendet, die Done Kriterien zu erstellen, die ein Output des "Priorisiertes Produkt Backlog erstellen" Prozesses sind. Eine User Story gilt als done, wenn sie dem Product Owner demonstriert und von diesem abgenommen wurde. Der Product Owner beurteilt die User Storys auf Grundlage der Done Kriterien und der Abnahmekriterien.

Ertragswertanalyse

Ertragswertanalyse analysiert die aktuelle Projektleistung gegen die geplante Leistung zu einem bestimmten Zeitpunkt. Es misst aktuelle Abweichungen im Projekt-Zeitplan und die Kostenleistung und prognostiziert die endgültigen Kosten auf der Grundlage der ermittelten aktuellen Leistung.

Aufwandsabgeschätzte Aufgabenliste

Die Aufwands-abgeschätzte Aufgabenliste ist eine Liste der Aufgaben im Zusammenhang mit den in einem Sprint enthaltenen zugesagten User Storys. Geschätzter Aufwand wird in den von dem Team vereinbarten Abschätzkriterien ausgedrückt. Die aufwandsabgeschätzte Aufgabenliste wird vom Scrum Team während der Sprint Planungs Meetings zur Erstellung des Sprint Backlogs und des Sprint Burndown Charts verwendet.

Empirische Prozesssteuerung

Ein empirisches Prozesssteuerungs-Modell hilft dabei, Entscheidungen zu treffen, die auf Beobachtungen und Experimenten und nicht auf detaillierter Vorausplanung basieren. Es stützt sich auf die drei zentralen Ideen der Transparenz, Kontrolle und Anpassung.

Epic(s)

Epic(s) werden in der Anfangsphase des Projekts geschrieben, wenn die meisten User Storys noch high-level sind oder Produktbeschreibungen und Anforderungen nur allgemein definiert sind. Epics sind große, unverfeinerte User Storys im Priorisierten Produkt Backlog

Schätzungsbereich

Schätzungen für Projekte sollten in Bereichen dargestellt werden. Genaue Zahlen können den Eindruck erwecken sehr akkurat zu sein, wenn sie es tatsächlich nicht sind. In der Tat sind Schätzungen per Definition eben nicht akkurat. Der Bereich der Schätzungen sollte auf dem Grad der Zuversicht beruhen, welches das Team in jede Schätzung hat.

Aufgaben Abschätzungs Prozess

Prozess, bei dem das Scrum Kern Team in einem Aufwands Abschätzungs Workshop den Aufwand schätzt, der zur Ausführung jeder Aufgabe aus der Aufgabenliste notwendig ist. Das Ergebnis dieses Prozesses ergibt die aufwandsabgeschätzte Aufgabenliste.

Abschätzungskriterien

Primäres Ziel beim Einsatz von Abschätzungskriterien ist es, relative Schätzwerte beizubehalten und die Notwendigkeit für eine erneute Abschätzung zu minimieren. Abschätzungskriterien können auf unterschiedliche Art und Weise formuliert werden. Zwei weit verbreitete Beispiele sind Story Points und Idealzeit.

Erwarteter Geldwert

Dies ist eine Risiko-Bewertungs-Technik, bei der der potentielle finanzielle Einfluss eines Risikos bestimmt wird, basierend auf dem Erwarteten Geldwert (EMV). EMV wird berechnet durch Multiplikation der monetären Auswirkungen mit der Risikowahrscheinlichkeit, wie durch den Kunden angenähert

Forscher—Käufer—Urlauber—Gefangener (ESVP)

Dies ist eine Übung, die zu Beginn des Sprint Rueckblick Meetings durchgeführt wird, um die Denkweise der Teilnehmer besser zu verstehen und den Ton für die Besprechung anzugeben. Teilnehmer werden gebeten, anonym anzugeben, was ihre Einstellung am Besten repräsentiert.

Externe Abhängigkeiten

Externe Abhängigkeiten sind solche, die im Zusammenhang mit Aufgaben, Aktivitäten oder Produkten stehen, die nicht zum Arbeitsumfang des Scrum-Teams gehören, aber erforderlich sind, um eine Aufgabe im Projekt abzuschließen, oder ein zu lieferndes Ergebnis des Projekts zu erzeugen. Externe Abhängigkeiten sind in der Regel außerhalb der Kontrolle des Scrum-Teams.

Faust von Fünf

Unter Faust von Fünf (Fist of Five) versteht man einen einfachen und schnellen Mechanismus, mit dem innerhalb einer Gruppe Konsens erreicht und Diskussionen vorangetrieben werden können. Nach einer anfänglichen Diskussion über einen bestimmten Vorschlag oder eine bevorstehende Entscheidung werden die einzelnen Scrum Teammitglieder aufgefordert, mithilfe ihrer Finger auf einer Skala von 1 bis 5 abzustimmen.

Fokusgruppen Meetings

In einer Fokusgruppe teilen individuelle Teilnehmer in einer moderierten Sitzung ihre Meinungen, Wahrnehmungen oder Bewertungen eines Produkts, Services oder gewünschten Ergebnisses mit. Die Mitglieder einer Fokusgruppe dürfen einander Fragen stellen oder um Aufklärung bestimmter Themen oder Konzepte bitten. Anhand von Fragen, konstruktiver Kritik und Feedback führen Fokusgruppen zu einem Produkt von besserer Qualität und tragen somit zur Erfüllung der Erwartungen der Anwender bei.

Scrum Team bilden

Die Scrum-Team-Mitglieder werden während dieses Prozesses identifiziert. Normalerweise hat der Product Owner die primäre Verantwortung für die Auswahl der Teammitglieder, aber er oder sie macht dies oft in Zusammenarbeit mit dem Scrum Master.

Einstiegs- und Findungsphase

Einstiegs- und Findungsphase ist die erste Stufe der Teambildung, die oft als eine unterhaltsame Phase angesehen wird, weil alles neu ist und das Team noch auf keine Schwierigkeiten mit dem Projekt gestoßen ist.

Vier Fragen pro Team

Eine Reihe von Frage, die in jedem Scrum der Scrums (SoS) Meeting gestellt wird. Jede/r Scrum Team-Repräsentant/in berichtet vom Fortschritt seines/ihrer Teams, in dem er/sie in der Regel vier spezifische Fragen beantwortet.

1. Woran hat mein Team seit des letzten Meetings gearbeitet?
2. Was wird mein Team bis zum nächsten Meeting tun?
3. Gibt es unfertige Arbeiten meines Teams, auf dessen Fertigstellung sich andere Teams verlassen haben?
4. Gibt es innerhalb meines Teams Vorhaben, die sich auf andere Teams auswirken könnten?

Lückenanalyse

Lückenanalyse ist eine Technik, die dazu verwendet wird, um den aktuellen, tatsächlichen Zustand mit einem gewünschten Zustand zu vergleichen und festzustellen, wie die Kluft zwischen ihnen überbrückt werden kann.

Priorisiertes Produkt Backlog verfeinern

"Priorisiertes Produkt Backlog verfeinern" ist ein Prozess, in dem das Priorisierte Produkt Backlog ständig aktualisiert und gepflegt wird.

Prozess zur Identifizierung von Scrum Master und Business Stakeholdern(n)

Prozess, in dem Scrum Master und Business Stakeholdern anhand spezieller Auswahlkriterien identifiziert werden.

Impediment

Unter Impediments versteht man sämtliche Hindernisse oder Hemmnisse, die die Produktivität des Scrum Teams verringern.

Implementierungsphase

Die Implementierungsphase beinhaltet Prozesse, die mit der Ausführung der Aufgaben und Aktivitäten zur Erstellung des Produkts eines Projekts in Verbindung stehen.

Vertrag mit Anreizen und Geldbussen

Dieser Vertrag basiert auf der Vereinbarung, dass der Lieferant mit einem finanziellen Anreiz belohnt wird, wenn die Produkte des Projekts rechtzeitig geliefert werden, aber Geldstrafen auf sich zieht, wenn die Lieferung zu spät kommt.

Vertrag zur inkrementalen Bereitstellung

Dieser Vertrag umfasst Prüfpunkte in regelmäßigen Abständen. Er hilft den Kunden oder Business Stakeholdern in regelmäßigen Abständen während der gesamten Projektlaufzeit an jedem Prüfpunkt Entscheidungen hinsichtlich der Produktentwicklung zu treffen. Der Kunde kann entweder die Entwicklung des Produkts akzeptieren, beschließen die Entwicklung des Produkts zu stoppen, oder Produkthanpassungen verlangen.

Indexkarten

Index- oder auch Storykarten werden verwendet, um User Storys im Verlauf des Projekts zu verfolgen. Sie erhöhen Anschaulichkeit und Transparenz und erleichtern ein frühzeitiges Aufdecken möglicher Probleme.

Initiierungs Phase

Diese Phase besteht aus den Prozessen zur Initiierung eines Projektes: "Projektvision erstellen", "Scrum Master und die Business Stakeholdern identifizieren", "Scrum Team bilden", "Epics entwickeln", "Priorisiertes Produkt Backlog entwickeln", und "Release Planung durchführen".

Inspektion

Inspektion bezieht sich auf die zur empirischen Prozesssteuerung notwendige Überwachung. Inspektion stellt außerdem sicher, dass die zu liefernden Ergebnisse des Projekts den Anforderungen entsprechen

Interne Abhängigkeiten

Interne Abhängigkeiten sind die Abhängigkeiten zwischen Aufgaben, Produkten oder Aktivitäten, die unter der Kontrolle des Scrum Teams sind und zum durch das Scrum Team auszuführenden Arbeitsumfang gehören.

Interner Zinsfuß (IRR)

Interner Zinsfuß (IRR) ist ein Zinssatz für eine Investition, bei dem der gegenwärtige Wert der Bargeldzuflüsse dem gegenwärtigen Wert der Bargeldabflüsse gleichgesetzt wird, um den Zinsfuß des Projekts zu ermitteln. Beim Vergleich von Projekten ist das mit einem höheren IRR in der Regel besser.

Angelegenheiten

Angelegenheiten sind allgemein Dinge, die gegenwärtig im Projekt geschehen und somit gewiss sind. Anders als bei einem Risiko gibt es dementsprechend keine Notwendigkeit, eine Wahrscheinlichkeitseinschätzung durchzuführen.

Iterative Lieferung

Iterative Lieferung ist die stufenweise Lieferung von Mehrwert für den Kunden.

JAD Sitzungen

Unter einer Joint Application Design (JAD) Sitzung versteht man eine Technik zur Erfassung von Anforderungen. Sie findet als stark strukturierter und moderierter Workshop statt, der den Erstellungsprozess der Projektvision beschleunigt, da er die Business Stakeholdern und andere Entscheidungsträger in die Lage versetzt, einen Konsens bzgl. des Umfangs, der Ziele und weiterer Spezifikationen des Projekts zu finden.

Joint Venture Vertrag

Diese Vertragsform wird im Allgemeinen verwendet, wenn zwei oder mehrere Parteien zu Partnern werden, um ein Projekt gemeinsam zu bewältigen. Alle am Projekt beteiligten Parteien erzielen eine Investitionsrendite, da die generierten Umsätze oder Erträge auf die Parteien aufgeteilt werden.

Kano-Analyse

Kano-Analyse wurde von Noriaki Kano (1984) entwickelt und betrifft die Klassifizierung von Features oder Anforderungen in vier Kategorien, die auf der Grundlage von Kundenpräferenzen definiert werden:

1. Begeisterungs Features
2. Zufriedenheits Features
3. Ablehnungs Features
4. Unerhebliche Features

Unterstützende Leader

Unterstützende Leader benutzen Zuhören, Empathie, Engagement und Einsicht, während sie Macht und Autorität mit den Teammitgliedern teilen. Dienende Leader sind Stewards, die Ergebnisse erzielen, indem sie sich auf die Bedürfnisse des Teams konzentrieren.

Sprint-Länge

Auf der Grundlage der unterschiedlichen Vorgaben einschließlich der geschäftlichen Anforderungen und dem Release-Plan beschließen der Product Owner und das Scrum Team die Länge der Sprints für das Projekt. Einmal festgelegt ist die Sprint-Länge des Projekts in der Regel unveränderlich.

Die Sprint-Länge beschreibt die festgelegte Dauer der Sprints für ein Projekt.

Risiken

Risiken umfassen alle ungewissen oder außerplanmäßigen Ereignisse, die das Projekt positiv oder negativ beeinflussen können.

Zwingende Abhängigkeiten

Diese Abhängigkeiten ergeben sich entweder aus der Natur der Arbeit, wie etwa eine physische Einschränkung, oder aus vertraglichen Verpflichtungen oder gesetzlichen Anforderungen.

Marktstudie

Der Begriff Marktstudie bezieht sich auf die organisierte Forschung, Sammlung, Gegenüberstellung und Analyse von Daten, die im Zusammenhang mit Kundenpräferenzen für Produkte stehen. Oft enthält die Studie umfangreiche Daten über Markttrends, Marktsegmentierung und Marketingprozesse.

Minimale Abnahmekriterien

Minimale Abnahmekriterien werden durch den Geschäftsbereich festgelegt. Sie sind Bestandteil der Abnahmekriterien für jede User Story für diesen Geschäftsbereich. Jede Funktionalität, die durch diesen Geschäftsbereich definiert wird, muss diese minimalen Abnahmekriterien erfüllen, wenn sie von dem entsprechenden Product Owner akzeptiert werden soll.

Gemilderte Risiken

Unter gemilderten Risiken versteht man die Risiken, die im Laufe des Projekts vom Scrum Team erfolgreich adressiert oder gemildert werden.

Monopoly Geld

Technik, bei der dem Kunden „Spiel-“ bzw. „Monopoly-Geld“ in der Höhe des tatsächlichen Projektbudgets mit der Bitte ausgehändigt wird, dieses an die in Betracht gezogenen User Storys zu verteilen. Auf diese Weise priorisiert der Kunde aufgrund seiner Zahlungsbereitschaft für jede User Story.

MoSCoW-Priorisierung

Der Begriff MoSCoW-Priorisierung leitet seinen Namen von den Anfangsbuchstaben der Formulierungen „Must have“, „Should have“, „Could have“ und „Won't have“ ab. Diese legen Prioritäten in absteigender Reihenfolge fest, wobei sich „Must have“ Features so definieren, dass das Produkt ohne sie wertlos ist, und „Won't have“ Features für jene Eigenschaften stehen, die zwar nett wären, jedoch nicht zwingend inbegriffen sein müssen.

Kapitalwert (NPV)

Kapitalwert (NPV) ist eine Methode zur Bestimmung des aktuellen Kapitalwerts eines zukünftigen finanziellen Vorteils, basierend auf einer angenommenen Inflationsrate oder einem angenommenen Zinssatz.

Nicht-zentrale Rollen

Nicht-zentrale Rollen sind solche Rollen, die nicht zwingend für das Scrum-Projekt erforderlich sind. Sie können Teammitglieder beinhalten, die Interesse am Projekt haben, aber keine formale Rolle im Projektteam spielen. Diese Rollen mögen eine Schnittstelle zum Team haben, sind jedoch nicht für den Erfolg des Projekts verantwortlich.

Regel- und Übereinkommensphase

Die dritte Stufe der Teambildung, in der das Team anfängt, sich zu entwickeln, seine internen Meinungsverschiedenheiten zu klären und Lösungen zur Zusammenarbeit zu finden. Diese Stufe wird als Phase der Anpassung angesehen.

Anzahl der Storys

Anzahl der Storys bezieht sich auf die Anzahl der User Storys, die als Teil eines einzigen Sprints geliefert werden. Sie kann als einfache Anzahl oder auch gewichtete Anzahl dargestellt werden.

Gelegenheiten

Risiken, die wahrscheinlich einen positiven Einfluss auf das Projekt haben, werden als Gelegenheiten bezeichnet.

Gelegenheitskosten

Gelegenheitskosten beziehen sich auf den Wert der nächstbesten Geschäfts-Option oder des nächstbesten Projektes, die/das zugunsten des ausgewählten Projekts verworfen wurde.

Organisatorische Bereitstellungsmethoden

Die Bereitstellungsmechanismen verschiedener Organisationen neigen dazu, unterschiedlich zu sein, abhängig von Industrie, Zielenwender und Positionierung. Abhängig vom gelieferten Produkt kann die Bereitstellung aus der Ferne erfolgen, aber auch den physischen Versand oder die physische Übergabe eines Artikels beinhalten.

Organisatorische Ressourcen-Matrix

Die organisatorische Ressourcen-Matrix ist eine hierarchische Darstellung einer Kombination aus einer funktionalen Organisationsstruktur und eine Projekt-Organisationsstruktur. Matrix Organisationen bringen Teammitglieder für ein Projekt aus unterschiedlichen funktionalen Abteilungen wie Informationstechnologie, Finanzen, Marketing, Vertrieb, Fertigung und anderen Abteilungen zusammen und bilden funktionsübergreifende Teams.

Paarweiser Vergleich

Paarweiser Vergleich ist eine Technik, bei der eine Liste aller User Storys im Priorisierten Produkt Backlog vorbereitet wird. Anschließend wird jede einzelne User Story mit den anderen User Storys verglichen, Story für Story. Jedes Mal, wenn zwei User Storys miteinander verglichen werden, wird entschieden, welche von beiden wichtiger ist. Durch diesen Prozess kann eine priorisierte Liste der User Storys generiert werden.

Pareto-Analyse

Diese Technik der Risikobewertung umfasst das Einordnen von Risiken nach Größenordnung. Es hilft dem Scrum-Team die Risiken in der Reihenfolge ihrer möglichen Auswirkungen auf das Projekt anzugehen.

PDCA/PDSA Zyklus

Der Plan-Do-Check-Act-Zyklus – auch bekannt als Deming- oder Shewhart-Zyklus – wurde von Dr. W. Edwards Deming, dem Vater der modernen Qualitätskontrolle, und Dr. Walter A. Shewhart entwickelt. Deming änderte Plan-Do-Check-Act später zu Plan-Do-Study-Act (PDSA) um, weil er der Meinung war, dass der Begriff „Study“ die Analyse gegenüber der einfachen Überprüfung („Check“) stärker hervorhob. Sowohl Scrum als auch der Deming-/Shewhart-/PDCA-Zyklus verstehen sich als iterative Methoden, die sich auf kontinuierliche Verbesserungen konzentrieren.

Arbeits- und Leistungsphase

Die abschließende Phase der Teambildung, in der das Team am Besten zusammenhält und auf höchstem Leistungsniveau arbeitet. Die Mitglieder haben ein leistungsfähiges Team von gleichberechtigten Profis gebildet, die beständig produktiv sind.

Personas

Personas sind sehr detaillierte fiktive Charaktere, repräsentativ für die Mehrheit der Anwender und auch anderer Business Stakeholdern, die das Endprodukt nicht unbedingt direkt anwenden. Personas werden erstellt, um die Bedürfnisse der Zielanwender-Basis (besser) zu identifizieren.

Pilotierungsplan

Ein Pilotierungsplan kann verwendet werden, um eine Pilot-Bereitstellung im Detail auszuarbeiten. Umfang und Ziele der Bereitstellung, Ziel-Anwenderbasis, Bereitstellungszeitplan, Übergangspläne, benötigte Anwendervorbereitung, Bewertungskriterien für die Bereitstellung und andere wichtige Elemente, die im Zusammenhang mit der Bereitstellung stehen, werden im Pilotierungsplan spezifiziert und den Business Stakeholdern mitgeteilt.

Planungs- und Schätzungsphase

Planungs- und Schätzungsphase besteht aus Prozessen zur Planung und Schätzung von Aufgaben. Dazu gehören die folgenden Prozesse: User Storys erstellen; User Storys genehmigen, abschätzen und zusagen; Aufgaben erstellen; Aufgaben abschätzen; Sprint Backlog aktualisieren

Planung für Mehrwert

Planung für Mehrwert bezieht sich auf das Rechtfertigen und Bestätigen des Projektwerts. Die Beweislast dafür, wie Mehrwert erzeugt wird, fällt auf die Business Stakeholdern (Sponsor, Kunden und/oder Anwender), während das Scrum-Team sich darauf konzentriert, was entwickelt wird.

Planungspoker

Planungspoker, auch Schätzungspoker genannt, ist ein Verfahren zur Abschätzung der relativen Größe der User Storys oder des benötigten Entwicklungsaufwands, das Gruppendenken und individuelles Denken in Einklang bringt.

Story Points für die Kostenabschätzung

Kostenabschätzung kann durch die Verwendung von relativen Einheiten (z.B. Aufwandsabschätzungen) anstelle von absoluten Einheiten (d.h. tatsächlich verursachte Kosten) erreicht werden. Zur Abschätzung der Kosten zur Einführung einer User Story kann das Scrum Team Story Points benutzen. Die Kosten, die für jede User Story abgeschätzt wurden, liegen dann in der Form von Story Points vor, nicht als finanzielle Einheiten.

Portfolio

Ein Portfolio ist eine Gruppe zusammengehöriger Programme. Ein Portfolio hat das Ziel, die in der Portfoliovisionserklärung definierten Geschäftsergebnisse zu liefern. Das Priorisierte Portfolio Backlog verkörpert die Priorisierten Produkt Backlogs für alle Programme im Portfolio.

Portfolio Product Owner

Der Portfolio Product Owner definiert die strategischen Ziele und Prioritäten für das Portfolio.

Portfolio Scrum Master

Der Portfolio Scrum Master löst Probleme, beseitigt Impediments und moderiert bzw. leitet Meetings für das Portfolio.

Priorisierung

Unter Priorisierung versteht man die Festlegung der Reihenfolge der Dinge und die Trennung dessen, was sofort getan wird, von dem, was später getan werden kann.

Priorisiertes Produkt Backlog

Das Priorisierte Produkt-Backlog ist ein einzelnes Anforderungsdokument. Es definiert den Projektumfang mittels einer priorisierten Liste der Eigenschaften des Produkts oder der Dienstleistung, die durch das Projekt geliefert werden sollen.

Wahrscheinlichkeits- Auswirkungsraster

Ein Raster, mit dem die Risiken bezüglich der Wahrscheinlichkeit ihres Eintretens und der potentiellen Auswirkungen auf die Projektziele bewertet werden. Im Allgemeinen werden jedem Risiko zwei unabhängige numerische Werte zugewiesen: einer für die Wahrscheinlichkeit und einer für die Auswirkung. Die zwei Werte werden dann multipliziert, um die Schwere des entsprechenden Risikos zu bestimmen. Das Ergebnis kann zur Priorisierung von Risiken benutzt werden.

Wahrscheinlichkeitsbäume

Mögliche Ereignisse werden in einem Diagramm mit einem Zweig für jedes mögliche Ergebnis der Ereignisse dargestellt. Die Wahrscheinlichkeit jedes Ergebnisses wird auf dem entsprechenden Zweig dargestellt. Diese Werte können zur Berechnung der Gesamtwirkung des Eintretens von Risiken in einem Projekt verwendet werden.

Produkt

Im SBOK® bezieht sich der Begriff „Produkt“ auf ein Produkt, einen Service oder eine andere Leistung, welche/r/s für den Kunden einen Mehrwert liefert.

Priorisiertes Produkt Backlog Review Meeting

Ein Produkt Backlog Review Meeting (auch als Priorisiertes Produkt Backlog Pflege-Session bezeichnet) ist ein formelles Meeting während des "Priorisiertes Produkt Backlog verfeinern" Prozesses, das dem Scrum Team hilft das Priorisierte Produkt Backlog zu überprüfen und darüber Konsens zu erreichen.

Product Owner

Der Product Owner ist für die Maximierung des Geschäftlichen Werts verantwortlich. Gleichwohl ist es seine/ihre Aufgabe, die Anforderungen des Kunden zu artikulieren und die geschäftliche Rechtfertigung des Projekts zu erhalten.

Programm

Ein Programm ist eine Gruppe zusammengehöriger Projekte. Ein Programm hat das Ziel, die in der Programmvisionserklärung definierten Geschäftsergebnisse zu liefern. Das Priorisierte Programm Backlog verkörpert die Priorisierten Produkt Backlogs für alle Projekte im Programm.

Programm und Portfolio Risiken

Risiken im Zusammenhang mit einem Portfolio oder einem Programm, die auch die Projekte beeinflussen, die Teil des jeweiligen Portfolios oder Programms sind.

Programm Product Owner

Der Programm Product Owner definiert die strategischen Ziele und Prioritäten für das Programm.

Programm Scrum Master

Der Programm Scrum Master löst Probleme, beseitigt Hindernisse und moderiert bzw. leitet Meetings für das Programm.

Projekt

Unter einem Projekt versteht man ein gemeinschaftliches Unternehmen zur Erstellung von neuen Produkten oder Services oder Ergebnissen, wie sie in der Project Visionserklärung definiert sind. Beschränkungen, denen Projekte in der Regel unterliegen, sind Zeit, Kosten, Lieferumfang, Qualität, Personen und die Fähigkeiten der Organisation(en).

Projektnutzen

Der Projektnutzen umfasst alle messbaren Verbesserungen in einem Produkt, einer Dienstleistung oder einem Ergebnis, die durch den erfolgreichen Abschluss eines Projekts zur Verfügung gestellt werden könnten.

Projektbudget

Das Projektbudget ist ein Finanzdokument, das die Kosten für Menschen, Materialien und andere Aufwendungen in einem Projekt beinhaltet. Das Projektbudget wird in der Regel durch den (die) Sponsor (en) abgezeichnet, um sicherzustellen, dass genügend Geldmittel zur Verfügung stehen.

Projektauftrag

Ein Projektauftrag ist eine offizielle Erklärung der gewünschten Ziele und Ergebnisse des Projekts. In vielen Unternehmen ist der Projektauftrag das Dokument, das das Projekt offiziell und formell genehmigt, und das Team schriftlich autorisiert, die Projektarbeit zu beginnen.

Projektkosten

Die Projektkosten sind Investitionen und andere Entwicklungskosten für ein Projekt

Projektbegründung

Projektbegründung beinhaltet alle Faktoren, die das Projekt erfordern, ob positiv oder negativ, gewählt oder nicht (z. B. ungenügende Kapazität zur Deckung des bestehenden und des prognostizierten Bedarfs, Abnahme der Kundenzufriedenheit, niedrige Erträge, gesetzliche Anforderungen u.s.w.)

Zeitpläne

Zeitpläne spiegeln die Länge oder Dauer eines Projekts wieder. Zeitpläne in Bezug auf den Business Case beinhalten auch das Zeitintervall, über das der Nutzen des Projekts realisiert wird.

Projektvisions-Meeting

Ein Projektvisions-Meeting ist ein Meeting zwischen Programm-Business Stakeholdern, Programm Product Owner, Programm Scrum Master und Chief Product Owner. Es hilft bei der Identifizierung des Geschäftskontexts, der geschäftlichen Anforderungen und der Erwartungen der Business Stakeholder mit dem Ziel der Entwicklung einer effektiven Projektvisionsaussage.

Projektvisionserklärung

Der Schlüssel-Output des "Projektvision erstellen" Prozesses ist eine gut strukturierte Projektvisionserklärung. Eine gute Projektvision erklärt die Geschäftsbedürfnisse und die an das Projekt gestellten Erwartungen, aber nicht, wie diese Bedürfnisse erreicht werden.

Vorgeschlagene Nicht-Funktionale Einträge für das Produkt Backlog

Nicht-funktionale Anforderungen sind in den frühen Phasen des Projekts nicht unbedingt vollständig definiert und können während des Sprint Review Meetings oder dem Sprint Retrospektions-Meeting auftauchen. Diese Einträge sollten zum Priorisierten Produkt Backlog hinzugefügt werden, wenn sie entdeckt werden.

Qualität

Qualität wird definiert als die Fähigkeit des fertigen Produkts oder der fertigen zu liefernden Ergebnisse, die Abnahmekriterien zu erfüllen und den vom Kunden erwarteten geschäftlichen Wert zu erzielen.

Qualitätssicherung

Qualitätssicherung bezieht sich auf die Bewertung von Prozessen und Standards, die Qualitätsmanagement in einem Projekt regeln, um sicherzustellen, dass sie weiterhin relevant sind. Qualitätssicherungsmaßnahmen werden als Teil der Arbeit durchgeführt.

Qualitätskontrolle

Die Qualitätskontrolle bezieht sich auf die Durchführung der geplanten Qualitätsmaßnahmen, die vom Scrum Team bei der Erstellung von potenziell versandfähigen zu liefernden Ergebnissen ausgeführt werden. Qualitätskontrolle beinhaltet auch aus jeder Gruppe an vervollständigten Aktivitäten zum Zweck der kontinuierlichen Verbesserung zu lernen.

Qualitätsmanagement

Qualitätsmanagement in Scrum ermöglicht es Kunden, sich über die Probleme im Projekt früh bewusst zu sein und zu erkennen, ob ein Projekt für sie funktioniert oder nicht. Qualitätsmanagement in Scrum wird durch drei zusammenhängende Aktivitäten erleichtert:

1. Qualitätsplanung
2. Qualitätskontrolle
3. Qualitätssicherung

Qualitätsplanung

Qualitätsplanung bezieht sich sowohl auf die Identifikation und Definition des aus einem Sprint und dem Projekt benötigten Produkts sowie dessen Abnahmekriterien, als auch alle zu befolgenden Entwicklungsmethoden und die Hauptverantwortlichkeiten der Scrum-Team-Mitglieder in Bezug auf Qualität.

Refactoring

Refactoring ist ein Begriff aus der Softwareentwicklung. Das Ziel dieser Technik ist, die Wartbarkeit des vorhandenen Codes zu verbessern und ihn einfacher, präziser und flexibler zu machen. Refactoring bedeutet das Design des vorliegenden Codes zu verbessern ohne das Verhalten des Codes zu verändern. Es beinhaltet Folgendes:

- Die Beseitigung von sich wiederholendem und überflüssigem Code
- Zerlegung von Methoden und Funktionen in kleinere Routinen
- Klare Definition von Variablen und Methodennamen
- Vereinfachung des Codedesigns

- Den Code besser verständlich und leichter anpassbar machen

Abgelehnte zu liefernde Ergebnisse

Abgelehnte zu liefernde Ergebnisse sind solche, die die definierten Abnahmekriterien nicht erfüllen. Eine Liste der abgelehnten zu liefernden Ergebnisse wird nach jedem Sprint Review Meeting gepflegt und aktualisiert.

Relative Priorisierungsrangliste

Relative Priorisierungsrangliste ist eine einfache Auflistung der User Storys nach Priorität. Es ist eine wirksame Methode zur Bestimmung der gewünschten User Storys für die jeweilige Iteration oder das jeweilige Release des Produkts oder der Dienstleistung.

Relative Größenbestimmung/Story Points

Story Points werden nicht nur zur Kostenschätzung eingesetzt, sondern auch zur Abschätzung der allgemeinen Größe einer User Story oder eines Features. Bei diesem Ansatz werden Story Points basierend auf einer allgemeinen Bewertung der Größe einer User Story unter Berücksichtigung von Risiko, Aufwand und Komplexitätsgrad zugeteilt.

Release-Inhalt

Dieser besteht aus wesentlichen Informationen zu den zu liefernden Ergebnissen, die dem Kunden-Support-Team helfen können.

Freigabemitteilung

Freigabemitteilungen sollten externe oder marktorientierte Lieferkriterien für das zu liefernde Produkt beinhalten.

Release Plan

Ein Release-Plan ist einer der Schlüsseloutputs des "Release Planung durchführen" Prozesses. Ein Release-Plan gibt an, welche zu liefernden Ergebnisse an die Kunden freigegeben werden sollen sowie die geplanten Intervalle und Termine für die Releases. Es gibt nicht notwendigerweise nach jedem Sprint ein Release.

Release Planungs-Sessions

Das Hauptziel der Release Planungs-Sessions ist es, einen Releaseplan zu erstellen und dem Scrum-Team einen Überblick über die Releases und den Lieferplan für das von ihm zu entwickelnde Produkt an die Hand zu geben, so dass es sich an den Erwartungen des Product Owners und der relevanten Business Stakeholdern ausrichten kann.

Release Priorisierungs Methoden

Release Priorisierungs Methoden werden für die Entwicklung eines Release-Plans verwendet. Diese Methoden sind Industrie- und Organisations-spezifisch und werden normalerweise durch das Senior Management in einer Organisation festgelegt.

Gelöste Probleme

In Scrum der Scrums Meetings haben Scrum Team-Mitglieder die Möglichkeit, offen Probleme zu besprechen, die Einfluss auf ihr Projekt haben. Diese zeitnahe Diskussion und Lösung von Problemen im Scrum der Scrums Meeting verbessert die Koordination zwischen den verschiedenen Scrum Teams erheblich und reduziert auch den Bedarf an Redesign und Nachbearbeitung.

Release Rückblick halten

In diesem Prozess, der das Projekt abschließt, kommen organisatorische Business Stakeholdern und die Mitglieder des Scrum Kernteams zusammen, um auf das Projekt zurückzublicken und Lehren aus dem Projekt zu identifizieren, zu dokumentieren und zu verinnerlichen. Oft führen diese Lehren zur Dokumentation von vereinbarten Massnahmen, die in zukünftigen Projekten umgesetzt werden.

Projectretrospektions Meeting

Das Projektretrospektions Meeting ist ein Treffen, um Ansätze festzulegen, wie die Teamzusammenarbeit und die Effektivität für zukünftige Projekte verbessert werden können. Es wird ebenfalls diskutiert, was positiv oder negativ war, und welche potentiellen Verbesserungsmöglichkeiten es gibt. Dieses Treffen ist nicht zeitlich begrenzt (time-boxed) und kann persönlich oder in einem virtuellen Format durchgeführt werden.

Sprint Rückblick halten

In diesem Prozess treffen sich Scrum Master und das Scrum Team, um über die Lehren zu diskutieren, die im Verlauf des Sprints gezogen wurden. Diese werden dokumentiert und können bei zukünftigen Sprints zur Anwendung kommen.

Sprint Rückblick Log(s)

Das Sprint Rückblick Log ist eine Aufzeichnung der Meinungen, Diskussionen und Aktionspunkte, die in einem Sprint Retrospektions-Meeting angesprochen wurden. Der Scrum Master kann dieses Log anhand des Inputs von den Mitgliedern des Scrum Kernteams erstellen.

Sprint Retrospektions-Meeting

Das Sprint Retrospektions-Meeting ist zeitlich auf 4 Stunden begrenzt (für einen einmonatigen Sprint) und wird als Teil des "Sprint Rückblick halten" Prozesses durchgeführt. Die Länge kann entsprechend der Länge des Sprints skaliert werden. Bei diesem Meeting kommt das Scrum-Team zusammen, um die vorherigen Sprints zu überprüfen und zu reflektieren hinsichtlich der Prozesse, denen es gefolgt ist, der eingesetzten

Hilfsmittel, Zusammenarbeit, Kommunikationsmechanismen und anderer für das Projekt relevanten Aspekte.

Investitionsrendite (ROI)

Investitionsrendite (ROI), wenn für die Projektrechtfertigung verwendet, beurteilt die erwarteten Nettoerträge, die aus einem Projekt gewonnen werden. Es wird berechnet, indem zunächst die zu erwartenden Kosten oder Investitionen in ein Projekt vom erwarteten Umsatz abgezogen werden. Dieser Nettogewinn wird dann durch die zu erwartenden Kosten dividiert, um die Ertragsrate zu bekommen.

Risiko

Risiko ist definiert als ein ungewisses Ereignis oder eine Reihe an Ereignissen, das / die die Ziele eines Projekts beeinflussen oder zu seinem Erfolg oder Versagen beitragen kann / können.

Risikobereitschaft

Risikobereitschaft bezieht sich darauf, wie viel Unsicherheit ein Business Stakeholdern oder eine Organisation bereit ist zu übernehmen.

Risikobewertung

Die Bewertung bezieht sich darauf, die identifizierten Risiken zu bewerten und abzuschätzen.

Risikoverhalten

Im Wesentlichen bestimmt Risikoverhalten der Business Stakeholdern, wieviel Risiko für den/die Business Stakeholdern akzeptabel ist. Risikoverhalten ist ein entscheidender Faktor, wenn Business Stakeholdern eine Entscheidung hinsichtlich Maßnahmen zur Minderung potenzieller negativer Risiken treffen.

Risikoscheu

Risikoscheu ist eine der Kategorien der Nutzenfunktion. Sie bezieht sich auf einen Business Stakeholdern, der nicht bereit ist, ein Risiko einzugehen, egal was der erwartete Nutzen oder die erwarteten Gelegenheiten sind.

Risikostrukturplan

In dieser Struktur werden die Risiken anhand ihrer Kategorien oder Gemeinsamkeiten gruppiert. Risiken können z.B. als finanziell, technisch oder sicherheitsrelevant kategorisiert werden.

Risiko Burndown Chart

Ein Diagramm, das die kumulative Schwere der Projekt Risiken auf der Zeitachse zeigt. Die Wahrscheinlichkeiten der verschiedenen Risiken werden übereinander gezeichnet, um das kumulative Risiko auf der y-Achse anzuzeigen. Die erstmalige Ermittlung und Bewertung der Risiken und die Erstellung des Risiko-Burndown-Chart werden frühzeitig im Projekt durchgeführt.

Risiko Checklisten

Risiko Checklisten enthalten die wichtigsten Punkte, auf die bei der Ermittlung von Risiken geachtet werden muss, häufig auftretende Risiken in Scrum Projekten, oder sogar Gruppen von Risiken, die durch das Team adressiert werden sollten.

Risikokommunikation

Risikokommunikation beinhaltet das Kommunizieren der Erkenntnisse aus den ersten vier Schritte des Risikomanagements an die entsprechenden Business Stakeholdern und die Ermittlung ihrer Auffassung in Bezug auf die ungewissen Ereignisse.

Risikoidentifikation

Risikoidentifikation ist ein wichtiger Schritt im Risikomanagement. Er beinhaltet die Anwendung verschiedener Techniken mit dem Ziel, alle potenziellen Risiken zu identifizieren.

Risiko Treffen

Risiken lassen sich vom Product Owner leichter priorisieren, indem eine Sitzung des Scrum Kernteams einberufen wird. Andere relevante Business Stakeholdern können optional zu dem Meeting eingeladen werden.

Risikominderung

Risikominderung ist ein wichtiger Schritt im Risikomanagement. Dieser Schritt beinhaltet die Entwicklung einer geeigneten Strategie, um mit einem Risiko umzugehen.

Risikoneutral

Risikoneutral ist eine der Kategorien der Nutzenfunktion, die sich auf einen Business Stakeholdern bezieht, der weder risikoscheu noch risikosuchend ist. Betroffene Entscheidungen sind nicht von der Unwägbarkeit des Ergebnisses beeinflusst. Wenn zwei mögliche Szenarien den gleichen Nutzen bringen, wird sich der risikoneutrale Business Stakeholdern nicht darum kümmern, ob ein Szenario risikoreicher ist, als das andere.

Risikopriorisierung

Risikopriorisierung ist ein wichtiger Schritt im Risikomanagement. Dieser Schritt beinhaltet die Priorisierung von Risiken, die für spezifische Maßnahmen in das priorisierte Produkt Backlog aufgenommen werden sollen.

Risiko Stimulierungs-Listen

Risiko Stimulierungs-Listen werden dazu benutzt, Gedanken in Bezug auf die potentiellen Ursachen von Risiken zu stimulieren. Es stehen Risiko Stimulierungs-Listen für verschiedene Industriezweige und Projekttypen zur Verfügung.

Risikosuchend

Risikosuchend ist einer der Kategorien der Nutzenfunktion, die sich auf einen Business Stakeholdern bezieht, der gewillt ist das Risiko zu akzeptieren, selbst wenn es nur eine knappe Erhöhung des Ertrags oder Nutzens für das Projekt liefert.

Risikoschwellwert

Risikoschwellwert bezieht sich auf den Grad, auf dem ein Risiko für die Organisation des Business Stakeholders akzeptabel ist. Ein Risiko liegt entweder über oder unter dem Risikoschwellwert. Liegt sie darunter, ist der Teilhaber oder die Organisation eher bereit das Risiko akzeptieren.

Risikotoleranz

Die Risikotoleranz gibt Grad, Menge oder Volumen an Risiko an, denen die Teilhaber standhalten werden.

Risikobasierter Spike

Risikobasierte Spikes (Spitzen) sind im Grunde Experimente, bei denen Forschung oder Prototyping einbezogen sind, um potentielle Risiken besser zu verstehen. Bei einem Spike wird eine intensive zwei bis dreitägige Übung (vorzugsweise zu Beginn eines Projekts bevor die Epics entwickelt sind oder das Priorisierte Produkt Backlog erstellt wird) durchgeführt, um dem Team zu helfen, die Unwägbarkeiten zu bestimmen, die das Projekt beeinflussen könnten.

Umfang

Der Umfang eines Projektes ist die Gesamtsumme aller Produkt Inkremente und der für die Entwicklung des Endproduktes erforderlichen Arbeiten.

Scrum Guidance Body

Der Scrum Guidance Body (SGB) ist eine optionale Rolle. Er besteht in der Regel aus einer Reihe von Dokumenten und / oder einer Gruppe von Experten, die in der Regel in die Festlegung der Ziele hinsichtlich Qualität, staatlicher Vorschriften, Sicherheit und anderer organisatorischer Schlüsselparameter einbezogen sind.

Fachkenntnis des Scrum Guidance Body

Fachkenntnis des Scrum Guidance Body bezieht sich auf die dokumentierten Regeln und Verordnungen, Entwicklungsanleitungen oder Standards, und beste Praktiken.

Scrum Master

Der Scrum Master erfüllt eine der Scrum Kernteam Rollen. Er oder sie unterstützt die Erschaffung der zu erbringenden Projektleistungen, verwaltet Risiken, Änderungen und Hindernisse bei der Durchführung täglicher Standups, Sprint Rückblick halten und anderen Scrum Prozessen.

Scrum der Scrums Meeting

Wenn Scrum für große Projekte skaliert werden muss, ist das Scrum der Scrums Meeting ein wichtiges Treffen, an dem Vertreter aller Teams teilnehmen. Dieses Meeting wird in der Regel durch den Chief Scrum Master organisiert und dient dem Fokus auf Koordinations- und Integrationsbereiche zwischen den einzelnen Scrum Teams. Es findet sowohl in im Vorfeld festgelegten Intervallen als auch auf Anforderung der Scrum Teams statt.

Scrum Team

Das Scrum Team ist eine der Scrum Kernrollen. Das Scrum Team arbeitet an der Erstellung der Projektergebnisse und wirkt an der Erzeugung von geschäftlichem Mehrwert für das Projekt und alle Business Stakeholdern mit.

Scrum Team Lessons Learned

Vom selbstorganisierten und ermächtigten Scrum Team wird erwartet, dass es aus in einem Sprint gemachten Fehlern Lehren zieht, die den Teams in zukünftigen Sprints dabei helfen, ihre Leistung zu verbessern.

Scrum Team Repräsentanten

Ein vom Team gewählter Repräsentant für Scrum der Scrums (SoS) Meetings. Die Auswahl basiert darauf, welches Mitglied aufgrund der jeweiligen Probleme und Umstände diese Rolle am Besten erfüllt.

Scrumboard

Scrumboard ist ein Werkzeug, das vom Scrum Team zur Planung und Verfolgung der Fortschritte in jedem Sprint benutzt wird. Das Scrumboard enthält vier Spalten, mit denen der Fortschritt der geschätzten Aufgaben für den Sprint angegeben wird: eine Spalte "Zu erledigen" für Aufgaben die noch nicht begonnen haben, eine Spalte "In Arbeit" für Aufgaben die begonnen haben, aber noch nicht abgeschlossen sind, eine Spalte "Im Test" für Aufgaben die erledigt wurden, sich aber gegenwärtig im Test befinden und eine Spalte "erledigt" für die Aufgaben, die erledigt wurden und erfolgreich getestet sind.

Selbstorganisation

Scrum geht davon aus, dass Mitarbeiter Eigenmotivation besitzen und nach größerer Verantwortung streben. Dementsprechend erzeugen sie selbstbestimmt grösseren Mehrwert.

Unterstützender Leader

Unterstützende Leader benutzen Zuhören, Empathie, Engagement und Einsicht, während sie Macht und Autorität mit den Teammitgliedern teilen. Dienende Leader sind Stewards, die Ergebnisse erzielen, indem sie sich auf die Bedürfnisse des Teams konzentrieren.

Zu liefernde Ergebnisse ausliefern

In diesem Prozess werden Akzeptierte zu liefernde Ergebnisse ausgeliefert oder an den zuständigen Business Stakeholdern weitergeleitet. Eine formale Vereinbarung über Funktionierende zu liefernde Ergebnisse dokumentiert den erfolgreichen Abschluss des Sprints.

Einfache Schemata

Einfache Schemata involvieren die Kennzeichnung von Gegenständen als "1", "2", "3" oder "Hoch", "Mittel" und "Niedrig" und so weiter. Obwohl dies eine einfache und unkomplizierte Vorgehensweise ist, kann sie problematisch werden, da es oft eine Tendenz gibt, alles als Priorität "1" oder "Hoch" zu kennzeichnen.

Fähigkeiten-Anforderungs Matrix

Die Fähigkeits-Anforderungs Matrix, auch bekannt als ein Kompetenz-Rahmen, wird zur Beurteilung von Qualifikationslücken und Ausbildungsanforderungen für Teammitglieder benutzt. Eine Fähigkeits-Matrix erfasst die Kenntnisse, Fähigkeiten der Teammitglieder sowie deren Interesse an der Anwendung dieser Kenntnisse und Fähigkeiten in einem Projekt. Mit dieser Matrix, können die Organisations-Qualifikationslücken bei den Team-Mitgliedern bewertet werden und die Mitarbeiter erkennen, ob sie Weiterbildung in einem bestimmten Bereich oder Kompetenz benötigen.

Schnellboot

Schnellboot ist eine Technik, die verwendet werden kann, um die Sprint Retrospektions-Meetings durchzuführen. Team-Mitglieder spielen die Rolle der Besatzung eines Schnellboots. Das Boot muss eine Insel erreichen, die symbolisch für die Projektvision steht. Selbstklebende Notizzettel werden von den Teilnehmern zum Aufzeichnen von Motoren und Ankern benutzt. Motoren sind Dinge, die ihnen helfen, die Insel zu erreichen, während Anker Dinge sind, die sie dabei behindern die Insel zu erreichen. Diese Übung ist auf ein paar Minuten zeitlich begrenzt (time-boxed).

Sponsor

Der Sponsor ist eine individuelle Person oder eine Organisation, der Mittel oder Unterstützung für das Projekt bietet. Der Sponsor ist außerdem der Business Stakeholdern dem letztendlich alle Rechenschaft ablegen.

Sprint

Unter einem Sprint versteht man einen mit einem festen Zeitfenster (einer Time-Box) versehenen Zyklus mit einer Länge von ein bis sechs Wochen, während deren das Scrum-Team die Sprint Ergebnisse erarbeitet und erstellt.

Sprint Backlog

Das Sprint Backlog beschreibt die Liste der vom Scrum-Team im kommenden Sprint auszuführenden Aufgaben.

Sprint Burndown Chart

Beim Sprint Burndown Chart handelt es sich um eine Graphik, die den Umfang der im laufenden Sprint noch zu verrichtenden Arbeit darstellt.

Zu liefernde Ergebnisse des Sprints

Zu liefernde Ergebnisse des Sprints beziehen sich auf Produkt-Inkrementen oder zu erbringende Ergebnisse, die am Ende eines jeden Sprint abgeschlossen sind.

Sprint Planungs Meeting

Eine Sprint Planungsmeeting wird zu Beginn eines Sprints im Rahmen des "Sprint Backlog aktualisieren" Prozesses durchgeführt. Es ist zeitlich begrenzt (time-boxed) auf bis zu acht Stunden (für einen einmonatigen Sprint) und ist in zwei Teile gegliedert - Definition der Ziele und Abchätzung der Aufgaben.

Sprint Review Meeting

Das Sprint Review Meeting ist zeitlich begrenzt (time-boxed) auf vier Stunden (für einen einmonatigen Sprint) und kann entsprechend der Länge des Sprints skaliert werden. Während des Sprint Review Meetings präsentiert das Scrum-Team dem Product Owner die Ergebnisse des aktuellen Sprints. Der Product Owner kann die Ergebnisse akzeptieren oder ablehnen.

Sprint-Verfolgungs-Werkzeuge

Sprint-Verfolgungs-Werkzeuge werden verwendet, um den Fortschritt eines Sprints zu verfolgen und zu wissen, wo das Scrum-Team im Hinblick auf die Bewältigung der Aufgaben im Sprint Backlog steht. Es gibt eine ganze Reihe von Werkzeugen, die dazu verwendet werden können, die Arbeit in einem Sprint zu verfolgen. Eines der Häufigsten ist ein Scrumboard, auch als Aufgabentafel oder Fortschritts-Grafik bekannt.

Sprint Geschwindigkeit

Sprint-Geschwindigkeit ist die Geschwindigkeit, in der das Team die Arbeit in einem Sprint erledigen kann. Sie wird in der Regel in den gleichen Einheiten ausgedrückt, wie sie beim Schätzen verwendet werden, normalerweise Story Points oder ideale Zeit

Business Stakeholdern

Business Stakeholdern schließt als umfassender Begriff Kunden, Anwender und Sponsoren ein, die regelmäßig mit Product Owner, Scrum Master und Scrum Team interagieren, um Input zu liefern und die Kreation des mit dem Projekt verbundenen Produkts, Dienstleistung oder anderer Ergebnisse zu unterstützen.

Streit- und Auseinandersetzungsphase

Die zweite Stufe der Teambildung, in der das Team mit dem Versuch beginnt, das Werk zu vollenden. Jedoch können Machtkämpfe auftreten und es gibt oft Chaos oder Verwirrung unter den Teammitgliedern.

Story Mapping

Unter Story Mapping versteht man eine Technik, die einen visuellen Abriss des Produkts und dessen Leitkomponenten bietet. Von Jeff Patton (2005) als solches entworfen, wird Story Mapping in der Regel dazu verwendet, Produkt Roadmaps zu veranschaulichen. Story Maps zeigen die Reihenfolge der Iterationen der Produktentwicklung und zeigen einen Entwurf, welche Features im ersten, zweiten, dritten und anschließenden Releases enthalten sein werden.

Nachhaltiges Arbeitstempo

Nachhaltiges Arbeits-Tempo ist ein Tempo, das das Team komfortabel beibehalten kann. Ein nachhaltiges Arbeitstempo führt zu einer erhöhten Mitarbeiterzufriedenheit, Stabilität und erhöhten Schätzgenauigkeit. Das Alles führt letztlich zu mehr Kundenzufriedenheit.

SWOT-Analyse

SWOT (Englisch: Strengths, Weaknesses, Opportunities, Threats) ist ein strukturierter Ansatz zur Projektplanung, der hilfreich ist, um die Stärken, Schwächen, Gelegenheiten und Bedrohungen in Bezug auf ein Projekt zu bewerten. Diese Art der Analyse hilft bei der Identifizierung sowohl interner als auch externer Faktoren, die sich auf das Projekt auswirken könnten.

Zielkunden für das Release

Nicht jede Freigabe (Release) fasst alle Business Stakeholdern und Anwender ins Auge. Die Business Stakeholdern entscheiden sich vielleicht gewisse Freigaben auf eine Untermenge von Anwendern zu begrenzen. Der Freigabeplan spezifiziert die Zielkunden für die Freigabe.

Aufgaben Abschätzungs-Workshop

Aufgaben Abschätzungs-Workshops ermöglichen dem Scrum-Team die Abschätzung des Aufwands für die Durchführung einer Aufgabe oder einer Reihe von Aufgaben sowie des Personal-Aufwands und anderer Ressourcen, die zur Durchführung der Aufgaben innerhalb eines bestimmten Sprints benötigt werden.

Aufgabenliste

Hierbei handelt es sich um eine umfassende Liste, die alle Aufgaben enthält, denen sich das Scrum Team im aktuellen Sprint verpflichtet hat. Alle Aufgaben sind in dieser Liste beschrieben.

Aufgabenplanungs-Meeting

In einem Aufgabenplanungs-Meeting trifft sich das Scrum Team, um die Arbeit für den Sprint zu planen. Das Team bespricht die zugesagten User Storys an der Spitze des priorisierten Produkt Backlogs. Um sicherzustellen, dass die Gruppe beim Thema bleibt, sollte dieses Treffen zeitlich begrenzt (time-boxed) sein. Das Standardzeitfenster ist eine Begrenzung auf zwei Stunden pro Woche an Sprint-Länge.

Aufgabenorientierter Leader

Aufgabenorientierte Führer setzen Aufgabenerledigung und Termintreue durch.

Team-Building Plan

Da Scrum-Teams funktionsübergreifend sind, muss jedes Mitglied aktiv an allen Aspekten des Projekts teilnehmen. Der Scrum-Master sollte potenzielle Probleme, denen Teammitglieder ausgesetzt sein können, identifizieren und versuchen, sie mit dem Ziel eines effektiven Teams im Team Building Plan zu adressieren.

Teamkalender

Ein Teamkalender enthält Informationen über die Verfügbarkeit der Teammitglieder einschließlich der Informationen, die sich auf Mitarbeiter-Urlaub, -Abwesenheiten, wichtige Ereignisse und Feiertage beziehen.

Team Fachkenntnisse

Team Fachkenntnisse beziehen sich auf das Know-how der Scrum Team-Mitglieder, die User Storys und Aufgaben im Sprint Backlog gut genug zu verstehen, um die letztendlichen zu liefernden Ergebnisse zu erzeugen. Teamkompetenzen werden verwendet, um die Inputs zu bewerten, die benötigt werden, um die geplanten Arbeiten des Projekts auszuführen.

Technische Schuld

Technische Schuld (auch Design-Schuld oder Code Schuld genannt) kann entstehen, während das Team an der Schaffung der primären zu liefernden Ergebnisse des mit dem Projekt verbundenen (Software-) Produkts arbeitet. Es handelt sich dabei um den Teil der Arbeit, der vom Team als niedriger Priorität angesehen und gar nicht oder nur unvollständig ausgeführt wird. Technische Schuld sammelt sich an und muss irgendwann bezahlt werden.

X-Theorie

X-Theorie Leader sind der Annahme, dass Mitarbeiter von Natur aus unmotiviert sind und Arbeit wenn möglich vermeiden, wodurch ein autoritärer Führungsstil gerechtfertigt wird.

Y-Theorie

Y-Theorie Leader sind der Annahme, dass Mitarbeiter Eigenmotivation besitzen und nach größerer Verantwortung streben, weshalb die Y-Theorie mit einem partizipativeren Führungsstil einhergeht.

Gefahren

Gefahren sind Risiken, die das Projekt auf negative Weise beeinflussen könnten.

Drei Tägliche Fragen

Die Drei Täglichen Fragen werden beim vom Scrum Master organisierten täglichen Standup Meeting verwendet. Dabei stellt jedes Scrum Teammitglied Informationen in Form von Antworten auf drei konkrete Fragen zur Verfügung:

- Was habe ich gestern zu vervollständigt?
- Was werde ich heute vervollständigen?
- Welchen Hindernissen oder Behinderungen (falls vorhanden) stehe ich derzeit gegenüber?

Time-Boxing

Time-Boxing beschreibt die Einteilung in kurze Zeitfenster (time-boxes), binnen denen Arbeiten fertiggestellt werden müssen. Nach Ablauf des Zeitfensters (der Time-Box) nicht fertiggestellte Arbeiten werden in eine spätere Time-Box verschoben. Da Scrum Projekte einen gewissen Unsicherheitsfaktor mit sich bringen und von Natur aus dynamisch und anfällig für häufige Änderungen sind, bieten Time-Boxen die für sie wichtige Struktur.

Transparenz

Durch Transparenz wird jeglichem Betrachter die Wahrnehmung sämtlicher Facetten eines jeden Scrum Prozesses ermöglicht. Durch allumfassenden Informationsaustausch entsteht ein Umfeld mit einem hohen Maß an Vertrauen.

Nicht genehmigte Change Requests

Anfragen auf Änderungen werden in der Regel als Change Requests eingereicht. Change Requests bleiben ungenehmigt bis sie formell genehmigt werden

Aktualisiertes Programm Produkt Backlog

Ein Programm Produkt Backlog, das einer periodischen Pflege unterzogen wird, um Änderungen und neue Anforderungen zu integrieren.

Anwender

Anwender sind die Personen oder die Organisation, die Produkt, Service oder andere Ergebnisse des Projekts direkt benutzen. Ähnlich zu Kunden, können Anwender sowohl organisations-interne als auch -externe Anwender sein. In einigen Fällen können Kunden und Anwender derselbe sein.

Anwender-Gruppen Meetings

Anwender-Gruppen Meetings betreffen relevante Business Stakeholdern, in erster Linie Anwender oder Kunden des Produktes. Sie liefern dem Scrum-Kernteam mit Informationen aus erster Hand hinsichtlich der Erwartungen der Anwender. Dies hilft bei der Formulierung der Abnahmekriterien für das Produkt und liefert wertvolle Erkenntnisse für die Entwicklung von Epics

User Storys

User Storys befolgen eine spezielle, vordefinierte Struktur und dokumentieren auf vereinfachte Weise die Anforderungen und gewünschten Endbenutzerfunktionen. Die in User Storys geäußerten Anforderungen sind kurze, einfache und leicht verständliche Aussagen, die eine verstärkte Kommunikation zwischen den Business Stakeholdern und bessere Abschätzungen des Teams zur Folge haben.

User Story Abnahmekriterien

Jede User Story hat zugeordnete Abnahmekriterien. User Storys sind subjektiv. Die Abnahmekriterien liefern die erforderliche Objektivität um beim Sprint Review zu entscheiden, welche User Storys akzeptiert oder abgelehnt werden. Abnahmekriterien liefern Klarheit für das Team, was von einer User Story zu erwarten ist.

User Story Workshops

User Story Workshops werden im Rahmen des Prozesses "Epic (s) entwickeln" durchgeführt. Der Scrum Master moderiert diese Sitzungen. Das gesamte Scrum Kernteam ist beteiligt und manchmal ist es wünschenswert, andere Business Stakeholdern einzubeziehen.

User Story Schreibfachkenntnis

Basierend auf seiner oder ihrer Interaktion mit den Business Stakeholdern, eigenen betriebswirtschaftlichen Kenntnissen und Know-how sowie Inputs aus dem Team entwickelt der Product Owner User Storys, die das anfängliche Priorisierte Produkt Backlog für das Projekt bilden.

Nutzenfunktion

Nutzenfunktion ist ein Modell zur Messung der Risikopräferenz oder Einstellung zum Risiko der Business Stakeholdern. Es definiert den Level der Risikobereitschaft der Business Stakeholdern.

Value Stream Mapping

Value Stream Mapping verwendet Flussdiagramme, um den Fluss der für einen Prozess erforderlichen Schritte zu veranschaulichen. Diese Technik kann zur Optimierung eines Prozesses eingesetzt werden, indem sie hilft, nicht wertschöpfende Elemente zu identifizieren und zu eliminieren und die Effizienz zu steigern. Value Stream Mapping kann zur Optimierung von Scrum-Prozessen verwendet werden, um beispielsweise die Sprint-Geschwindigkeit zu verbessern.

Anbieter

Anbieter umfassen externe Personen oder Organisationen, die Produkte und Dienstleistungen bereitstellen, die nicht innerhalb der Kernkompetenzen der Projektorganisation sind.

Stimme des Kunden (VOC)

Sowohl explizite als auch implizite Anforderungen des Kunden, die vor der Konzeption eines Produkts oder eines Service verstanden werden müssen, können als Stimme des Kunden (VOC) bezeichnet werden. Der Product Owner repräsentiert die Stimme des Kunden.

War Room

Mit "War Room" wird Teamraum für alle Scrum-Team-Mitglieder beschrieben. Normalerweise ist er so konzipiert, dass Teammitglieder dort arbeiten, sich frei bewegen und außerdem leicht miteinander kommunizieren können, weil sie sich in unmittelbarer Nähe zueinander befinden.

Breitband Delphi Technik

Breitband-Delphi ist eine gruppenbasierte Technik zur Aufwandsabschätzung. Jedes einzelne Team Mitglied gibt anonym Schätzungen für jedes Feature ab. Diese ersten Schätzungen werden dann in einem Diagramm dargestellt. Das Team erörtert dann die Faktoren, die die jeweiligen Schätzungen beeinflusst haben und gehen dann in eine zweite Runde der Abschätzungen. Dieser Vorgang wird wiederholt, bis die individuellen Schätzungen nahe beieinander liegen und ein Konsens für die Abschätzung erreicht werden kann.

Funktionierende zu liefernde Ergebnisse

Dieser Output sind die letzten versandfähigen zu liefernden Ergebnisse, für die das Projekt sanktioniert wurde.

Vereinbarung über Funktionierende zu liefernde Ergebnisse

Zu liefernde Ergebnisse, die die Annahmekriterien erfüllen, erhalten geschäftliche Abnahme und Genehmigung durch den Kunden oder den Sponsor.

Der grundlegende Leitfaden für eine erfolgreiche Projekt-Durchführung mittels Scrum.

Der SBOK® wurde entwickelt als Standardleitfaden sowohl für Organisationen und professionelle Anwender, die Scrum implementieren möchten, als auch für diejenigen, die Scrum schon jetzt anwenden, aber notwendige Verbesserungen an Ihren Prozessen vornehmen möchten.

Der SBOK® zieht sein kombiniertes Wissen und seine Einblicke aus Tausenden von Projekten, quer durch verschiedene Organisationen und Industriezweige. Die Beiträge vieler Scrum Experten sowie Praktiker der Projektimplementierung sind bei seiner Entwicklung berücksichtigt worden. Der Fokus von Scrum auf wertgetriebene Lieferung hilft Scrum Teams, in einem Projekt so früh wie möglichen Ergebnisse zu liefern. Auf diese Weise wird die Kapitalrendite jener Gesellschaften, die Scrum als ihren bevorzugten Rahmen zur Projektdurchführung verwenden, deutlich verbessert. Darüber hinaus ist es durch die Verwendung von kurzen, iterativen Produkt-Entwicklungszyklen und häufiger Interaktion zwischen den Kunden und den Scrum Teams einfach, Änderungen des Projektumfangs vorzunehmen.

Der SBOK® kann sowohl von erfahrenen Praktikern von Scrum oder anderen Methoden der Produkt- oder Dienst-Entwicklung als Referenz und Leitfaden verwendet werden, als auch von Personen, die keine vorherige Erfahrung oder Kenntnis von Scrum oder anderen Methoden der Projekt-Durchführung haben.

Das erste Kapitel beschreibt den Zweck und Rahmen des SBOK® und liefert eine Einführung in die Schlüsselkonzepte von Scrum. Es enthält eine Zusammenfassung der Scrum Prinzipien, Scrum Aspekte und Scrum Prozesse. Das zweite Kapitel vertieft die sechs Scrum Prinzipien, welche die Grundlage der Scrum Rahmenstruktur bilden. Vom dritten bis zum siebten Kapitel werden die fünf Scrum Aspekte ausgearbeitet, die bei allen Projekten angesprochen werden müssen - Organisation, geschäftliche Rechtfertigung, Qualität, Änderung und Risiko. Kapitel acht bis zwölf decken die 19 grundlegenden Scrum Prozesse ab, die bei einem Scrum Projekt zum Einsatz kommen. Diese Prozesse sind Teil der fünf Scrum Phasen: Initiierung, Planung und Abschätzung, Implementierung, Review und Rückblick, und Freigabe. Es werden sowohl Details der zugehörigen Inputs und Outputs jedes Prozesses beschrieben, als auch die verschiedenen Hilfsmittel, die jeweils verwendet werden können.

Diese vierte Ausgabe des SBOK® Guide ergänzt das kollektive Wissen über das Scrum-Framework miterweiterte Inhalte in Bezug auf die Skalierung von Scrum für große Projekte und die Skalierung von Scrum für Unternehmen in den Kapiteln 13 bzw. 14.

Obwohl der SBOK® ein sehr umfassendes Nachschlagewerk für Scrum ist, ist sein Inhalt so organisiert, dass, unabhängig von der vorherigen Scrum Kenntnis des jeweiligen Lesers, das Nachschlagen von Themen einfach und das Lesen unterhaltsam sein sollte.

