

A Guide to the
SCRUM BODY OF KNOWLEDGE
(SBOK™ GUIDE)

Ausgabe 3rd

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™)**

Ausgabe 3rd

Ein umfassender Leitfaden zur Projektdurchführung mittels Scrum

©2016 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 2016

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

Avondale, Arizona 85392 USA

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: 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 Neuauflage 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, gezogene 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 <i>SBOK™</i>	5
1.4	Rahmenstruktur des <i>SBOK™</i>	6
1.4.1	Wie wende ich den <i>SBOK™</i> an?	7
1.4.2	Scrum Prinzipien	8
1.4.3	Scrum Aspekte.....	10
1.4.4	Scrum Prozesse.....	15
1.5	Scrum versus traditionelles Projektmanagement.....	20
2.	PRINZIPIEN	21
2.1	Einführung.....	21
2.2	Leitfaden für Kernrollen.....	22
2.3	Empirische Prozesssteuerung	22
2.3.1	Transparenz	22
2.3.2	Inspektion	24
2.3.3	Adaptierung.....	24
2.4	Selbstorganisation	27
2.4.1	Benefits der Selbstorganisation	27
2.5	Kollaboration	29
2.5.1	Vorteile der Kollaboration bei Scrum Projekten	29
2.5.2	Die Bedeutung von Colocation für Kollaboration	31
2.6	Wertbasierte Priorisierung	31
2.7	Time-boxing	33
2.7.1	Scrum Time-boxes	34
2.8	Iterative Entwicklung	36
2.9	Scrum versus zu traditionelles Projektmanagement	38

3.	ORGANISATION.....	39
3.1	Einleitung.....	39
3.2	Leitfaden für die Kernrollen.....	40
3.3	Scrum Projekt Rollen.....	40
3.3.1	Kernrollen.....	41
3.3.2	Nicht-zentrale Rollen.....	42
3.4	Product Owner.....	44
3.4.1	Stimme des Kunden (VOC).....	45
3.4.2	Chief Product Owner.....	45
3.5	Scrum Master.....	47
3.5.1	Chief Scrum Master.....	48
3.6	Scrum Team.....	51
3.6.1	Wahl der Mitarbeiter.....	53
3.6.2	Die Größe des Scrum Teams.....	53
3.7	Scrum in Projekten, Programmen und Portfolios.....	54
3.7.1	Definition von Projekt, Programm und Portfolio.....	54
3.7.2	Scrum in Projekten.....	Error! Bookmark not defined.
3.7.3	Scrum in Programmen und Portfolios.....	Error! Bookmark not defined.
3.7.4	Beteiligung der Stakeholder erhalten.....	56
3.8	Zusammenfassung der Verantwortlichkeiten.....	57
3.9	Scrum versus traditionelles Projekt Management.....	58
3.10	Populäre Theorien aus dem Personalbereich und deren Relevanz in Scrum.....	59
3.10.1	Modell der Gruppendynamik nach Tuckman.....	59
3.10.2	Konfliktmanagement.....	60
3.10.3	Konfliktmanagement Techniken.....	60
3.10.4	Führungsstile.....	62
3.10.5	Hierarchie der Bedürfnisse Theorie nach Maslow.....	64
3.10.6	X-Theorie und Y-Theorie.....	65
4.	GESCHÄFTLICHE RECHTFERTIGUNG.....	66
4.1	Einleitung.....	66
4.2	Leitfaden für die Kernrollen.....	67

4.3	Wert-getriebene Lieferung	67
4.3.1	Verantwortlichkeiten des Product Owners in der Geschäftlichen Rechtfertigung.	69
4.3.2	Verantwortlichkeiten der anderen Scrum Rollen in der Geschäftlichen Rechtfertigung	70
4.4	Die Wichtigkeit der Geschäftlichen Rechtfertigung	71
4.4.1	Faktoren um Geschäftliche Rechtfertigung zu bestimmen.....	71
4.4.2	Geschäftliche Rechtfertigung und der Lebenszyklus des Projekts.....	72
4.5	Techniken der Geschäftlichen Rechtfertigung	74
4.5.1	Einschätzung des Projekt Mehrwerts.....	74
4.5.2	Planung für Mehrwert	76
4.5.3	Relative Priorisierungsrangliste.....	78
4.5.4	Story Mapping	78
4.6	Kontinuierliche Wertrechtfertigung	78
4.6.1	Ertragswertanalyse.....	79
4.6.2	Kumulatives Flussdiagramm (CFD)	81
4.7	Bestätigen der Ertragsrealisierung.....	82
4.7.1	Prototypen, Simulationen und Vorführungen.....	82
4.8	Zusammenfassung der Verantwortlichkeiten	83
4.9	Scrum versus traditionelles Projekt Management.....	84
5.	QUALITÄT.....	85
5.1	Einleitung	85
5.2	Leitfaden für die Kernrollen.....	86
5.3	Qualitätsdefinition	86
5.3.1	Qualität und Umfang	87
5.3.2	Qualität und Geschäftswert	87
5.4	Abnahmekriterien und das Priorisierte Produkt Backlog.....	88
5.4.1	Abnahmekriterien schreiben.....	90
5.4.2	Minimale Abnahmekriterien	Error! Bookmark not defined.
5.4.3	Definition von Done.....	90
5.4.4	Akzeptieren oder Ablehnen eines Priorisierten Produkt Backlog Artikels.....	91
5.5	Qualitätsmanagement in Scrum.....	92

5.5.1	Qualitätsplanung	94
5.5.2	Qualitätskontrolle und Qualitätssicherung	95
5.5.3	Plan-Do-Check-Act (PDCA) Zyklus.....	96
5.6	Zusammenfassung der Verantwortlichkeiten	98
5.7	Scrum versus traditionelles Projekt Management	99
6.	Änderungen.....	101
6.1	Einleitung	101
6.2	Leitfaden für die Kernrollen	102
6.3	Überblick.....	102
6.3.1	Nicht genehmigte und genehmigte Change Requests	104
6.4	Änderungen in Scrum	106
6.4.1	Das Gleichgewicht an Flexibilität und Stabilität halten.....	106
6.4.2	Flexibilität erreichen.....	106
6.5	Änderungen integrieren	112
6.5.1	Änderungen an einem Sprint	113
6.6	Änderungen in Programmen und Portfolios	118
6.6.1	Im Portfolio.....	118
6.6.2	Im Programm.....	118
6.7	Zusammenfassung der Verantwortlichkeiten	121
6.8	Scrum versus traditionelles Projekt Management	122
7.	RISIKO	123
7.1	Einführung.....	123
7.2	Leitfaden für Kernrollen.....	124
7.3	Was ist Risiko?	124
7.3.1	Der Unterschied zwischen Risiken und Problemen.....	125
7.3.2	Risikoverhalten	125
7.4	Verfahrensweise beim Risikomanagement.....	126
7.4.1	Risikoidentifikation	126
7.4.2	Risikobewertung.....	127
7.4.3	Risikopriorisierung.....	131
7.4.4	Risikominderung.....	132

7.4.5	Risikokommunikation	133
7.5	Minimierung von Risiken mit Hilfe von Scrum	135
7.6	Risiken bei Portfolios und Programmen	135
7.6.1	Bei Portfolios	136
7.6.2	Bei Programmen	136
7.7	Zusammenfassung der Verantwortlichkeiten	138
7.8	Scrum versus traditionelles Projektmanagement	139
8.	INITIIERUNG	141
8.1	Projektvision Erstellen	145
8.1.1	Inputs	147
8.1.2	Hilfsmittel	150
8.1.3	Outputs	151
8.2	Scrum Master und Stakeholder Identifizieren	153
8.2.1	Inputs	155
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	169
8.4.2	Hilfsmittel	173
8.4.3	Outputs	175
8.5	Priorisiertes Produkt Backlog Erstellen	178
8.5.1	Inputs	180
8.5.2	Hilfsmittel	181
8.5.3	Outputs	183
8.6	Release-Planung Durchführen	186
8.6.1	Inputs	188
8.6.2	Hilfsmittel	189

8.6.3	Outputs.....	190
8.7	Datenflussdiagramm der Phase.....	192
9.	PLANEN UND ABSCHÄTZEN	193
9.1	User Storys erstellen.....	197
9.1.1	Inputs.....	199
9.1.2	Hilfsmittel	200
9.1.3	Outputs.....	201
9.2	User Storys Genehmigen, Abschätzen und Zusagen	204
9.2.1	Inputs.....	206
9.2.2	Hilfsmittel	206
9.2.3	Outputs.....	209
9.3	Aufgaben erstellen	210
9.3.1	Inputs.....	211
9.3.2	Hilfsmittel	212
9.3.3	Outputs.....	212
9.4	Aufgaben Abschätzen.....	213
Inputs.....		215
9.4.2	Hilfsmittel	215
9.4.3	Outputs.....	217
9.5	Sprint Backlog erstellen	218
9.5.1	Inputs.....	224
9.5.2	Hilfsmittel	Error! Bookmark not defined.
9.5.3	Outputs.....	Error! Bookmark not defined.
9.6	Phasen Datenflussdiagramm	227
10.	IMPLEMENTIEREN	228
10.1	Zu liefernde Ergebnisse erzeugen	232
10.1.1	Inputs.....	234
10.1.2	Hilfsmittel	236
10.1.3	Outputs.....	237
10.2	Tägliches Standup durchführen	239
10.2.1	Inputs.....	241

10.2.2	Hilfsmittel	242
10.2.3	Outputs	243
10.3	Priorisiertes Produkt Backlog pflegen	245
1.	Scrum Kernteam*	245
2.	Priorisiertes Produkt Backlog*	245
3.	Abgelehnte zu liefernde Ergebnisse.....	245
4.	Genehmigte Change Requests	245
5.	Nicht genehmigte Change Requests.....	245
6.	Identifizierte Risiken	245
7.	Aktualisiertes Programm Produkt Backlog	245
8.	Sprint Ruckblicks Log(s)	245
9.	Abhängigkeiten	245
10.	Release-Plan	245
11.	Empfehlungen des Scrum Guidance Body	245
1.	Priorisiertes Produkt Backlog Review Meetings*	245
2.	Kommunikationstechniken	245
3.	Andere Priorisiertes Produkt Backlog Pflorgetechniken	245
1.	Aktualisiertes Priorisiertes Produkt Backlog*	245
2.	Aktualisierter Release-Plan	245
10.3.1	Inputs	247
10.3.2	Hilfsmittel	249
10.3.3	Outputs	250
10.4	Phasen Datenflussdiagramm	251
11.	Review und Rückblick	253
11.1	Zum Scrum der Scrum zusammenkommen	Error! Bookmark not defined.
11.1.1	Inputs	Error! Bookmark not defined.
11.1.2	Hilfsmittel	Error! Bookmark not defined.
11.1.3	Outputs	Error! Bookmark not defined.
11.2	Sprint Vorführen und Validieren	257
11.2.1	Inputs	259
11.2.2	Hilfsmittel	260

11.2.3	Outputs.....	261
11.3	Sprint Rückblick halten	262
11.3.1	Inputs.....	263
11.3.2	Hilfsmittel	263
11.3.3	Outputs.....	265
11.4	Datenflussdiagramm der Phase.....	268
12.	RELEASE	269
12.1	Zu liefernde Ergebnisse ausliefern.....	272
1.	Organisatorische Bereitstellungsmethoden*	272
2.	Kommunikationsplan.....	272
12.1.1	Inputs.....	273
12.1.2	Hilfsmittel	275
12.1.3	Outputs.....	275
12.2	Projekt Rückblick halten.....	277
1.	Scrum Kernteam(s)*.....	277
2.	Chief Scrum Master.....	277
3.	Chief Product Owner.....	277
4.	Stakeholder	277
5.	Empfehlungen des Scrum Guidance Body.....	277
1.	Projekt Rückblick- Meeting*	277
2.	Andere Hilfsmittel, um einen Projekt Rückblick zu halten.....	277
3.	Fachkenntnis des Scrum Guidance Body	277
1.	Vereinbarte Verbesserungsmaßnahmen*	277
2.	Zugeordnete Aktionspunkte und Fertigstellungstermine*	277
3.	Vorgeschlagene Nicht-Funktionale Einträge für das Programm Produkt Backlog und das Priorisierte Produkt Backlog	277
4.	Aktualisierte Empfehlungen des Scrum Guidance Body	277
12.2.1	Inputs.....	278
12.2.2	Hilfsmittel	279
12.2.3	Outputs.....	280
12.3	Phasen Datenflussdiagramm	281

ANHANG A. EIN AGILE ÜBERBLICK	337
ANHANG B. AUTOREN UND REZENSENTEN DES SBOK™	347
REFERENZEN	349
Glossar	350

Abbildungsverzeichnis

ABBILDUNG 1-1: SCRUM FLUSSDIAGRAM.....	ERROR! BOOKMARK NOT DEFINED.
ABBILDUNG 1-2: SBOK™ RAHMENSTRUKTUR	ERROR! BOOKMARK NOT DEFINED.
ABBILDUNG 1-3: SCRUM PRINZIPIEN.....	ERROR! BOOKMARK NOT DEFINED.
ABBILDUNG 1-4: ORGANISATION IN SCRUM	ERROR! BOOKMARK NOT DEFINED.
ABBILDUNG 2-1: TRANSPARENZ BEI SCRUM.....	ERROR! BOOKMARK NOT DEFINED.
ABBILDUNG 2-2: INSPEKTION BEI SCRUM	24
ABBILDUNG 2-3: ADAPTIERUNG IN SCRUM	25
ABBILDUNG 2-4: PROBLEME BEIM WASSERFALL-PROJEKTMANAGEMENT	26
ABBILDUNG 2-5: ZIELE EINES SICH SELBST ORGANISIERENDEN TEAMS.....	28
ABBILDUNG 2-6: VORTEILE DER KOLLABORATION BEI EINEM SCRUM-PROJEKT.....	30
ABBILDUNG 2-7: WERTBASIERTE PRIORISIERUNG	33
ABBILDUNG 2-8: TIME-BOXING-SITUATIONEN BEI SCRUM MEETINGS	35
ABBILDUNG 2-9: SCRUM VS. TRADITIONELLES WASSERFALL MODELL	37
ABBILDUNG 3-1: SCRUM ROLLEN—ÜBERBLICK.....	ERROR! BOOKMARK NOT DEFINED.
ABBILDUNG 3-2: GESTELLTE FRAGEN WÄHREND EINES SCRUM DER SCRUMS MEETING	ERROR! BOOKMARK NOT DEFINED.
ABBILDUNG 3-3: WÜNSCHENSWERTE EIGENSCHAFTEN FÜR DIE SCRUM KERNROLLEN	ERROR! BOOKMARK NOT DEFINED.
ABBILDUNG 3-4: SCRUM DER SCRUMS (SOS) MEETING.....	332
ABBILDUNG 3-5: SCRUM QUER DURCH DIE ORGANISATION FÜR PROJEKTE, PROGRAMME UND PORTFOLIOS	55
ABBILDUNG 3-6: PHASEN DER GRUPPENENTWICKLUNG NACH TUCKMAN	59
ABBILDUNG 3-7: HIERARCHIE DER BEDÜRFNISSE THEORIE NACH MASLOW	64
ABBILDUNG 4-1: LIEFERUNG VON MEHRWERT IN SCRUM VERSUS TRADITIONELLEN PROJEKTEN	69
ABBILDUNG 4-2: HIERARCHIE DER VERANTWORTLICHKEIT DER GESCHÄFTLICHEN RECHTFERTIGUNG	70
ABBILDUNG 4-3: GESCHÄFTLICHE RECHTFERTIGUNG UND DER PROJEKT LEBENSZYKLUS.....	73
ABBILDUNG 4-4: KANO-ANALYSE	77
ABBILDUNG 4-5: BEISPIEL KUMULATIVES FLUSSDIAGRAMM (CFD)	81
ABBILDUNG 5-1: PROJEKT-INKREMENT FLUSSDIAGRAMM.....	89
ABBILDUNG 5-2: PDCA ZYKLUS IN SCRUM.....	97
ABBILDUNG 6-1: BEISPIEL GENEHMIGTER ÄNDERUNGSPROZESS.....	105
ABBILDUNG 6-2: AKTUALISIEREN DES PRIORISIERTEN PRODUKT BACKLOGS MIT GENEHMIGTEN ÄNDERUNGEN	105
ABBILDUNG 6-3: SCRUM MERKMALE UM FLEXIBILITÄT ZU ERREICHEN	107
ABBILDUNG 6-4: BEWEGGRÜNDE DER STAKEHOLDER, ÄNDERUNGEN ANZUFragen.....	108
ABBILDUNG 6-5: BEWEGGRÜNDE DES SCRUM KERNTAMS, ÄNDERUNGEN ANZUFragen.....	109
ABBILDUNG 6-6: INTEGRATION VON ÄNDERUNG IN SCRUM.....	113
ABBILDUNG 6-7: AUSWIRKUNG DER ERWARTETEN ÄNDERUNGEN AUF DIE SPRINT-LÄNGE	115
ABBILDUNG 6-8: ÄNDERUNGEN IM PORTFOLIO UND PROGRAMM EINBINDEN	120
ABBILDUNG 7-1: BEISPIEL FÜR EINEN WAHRSCHEINLICHKEITSBAUM	128
ABBILDUNG 7-2: BEISPIEL EINES PARENTO-DIAGRAMMS.....	129
ABBILDUNG 7-3: BEISPIEL EINES WAHRSCHEINLICHKEITS- / AUSWIRKUNGSRASTERS	130
ABBILDUNG 7-4: ABLAUF DER RISIKOPRIORISIERUNG.....	132
ABBILDUNG 7-5: BEISPIEL EINES RISIKO-BURNDOWN-CHARTS	134
ABBILDUNG 7-6: HANDHABUNG VON RISIKEN IN PORTFOLIOS UND PROGRAMMEN.....	137
ABBILDUNG 8-1: ÜBERBLICK ÜBER DIE INITIIERUNGSPHASE	143
ABBILDUNG 8-2: ÜBERBLICK ÜBER DIE INITIIERUNGSPHASE (KERNPUNKTE).....	144

ABBILDUNG 8-3: PROJEKTVISION ERSTELLEN—INPUTS, HILFSMITTEL UND OUTPUTS	145
ABBILDUNG 8-4: PROJEKTVISION ERSTELLEN—DATENFLUSSDIAGRAMM	146
ABBILDUNG 8-5: DER VORGANG DER LÜCKENANALYSE	151
ABBILDUNG 8-6: SCRUM MASTER UND STAKEHOLDER IDENTIFIZIEREN—INPUTS, HILFSMITTEL UND OUTPUTS.....	153
ABBILDUNG 8-7: SCRUM MASTER UND STAKEHOLDER IDENTIFIZIEREN—DATENFLUSSDIAGRAMM	154
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	178
ABBILDUNG 8-13: ERSTELLEN EINES PRIORISIERTEN PRODUKT BACKLOGS: DATENFLUSSDIAGRAMM	179
ABBILDUNG 8-14: RELEASE-PLANUNG DURCHFÜHREN-INPUTS, HILFSMITTEL UND OUTPUTS.....	186
ABBILDUNG 8-15: RELEASE-PLANUNG DURCHFÜHREN—DATENFLUSSDIAGRAMM.....	187
ABBILDUNG 8-16: INITIIERUNGSPHASE-DATENFLUSSDIAGRAMM	192
ABBILDUNG 9-1: PLANUNGS- UND ABSCHÄTZUNGSÜBERSICHT	195
ABBILDUNG 9-2: PLANUNGS- UND ABSCHÄTZUNGSÜBERSICHT (KERNPUNKTE).....	196
ABBILDUNG 9-3: USER STORIES ERSTELLEN—INPUTS, HILFSMITTEL UND OUTPUTS.....	197
ABBILDUNG 9-4: USER STORIES ERSTELLEN—DATENFLUSSDIAGRAMM	198
ABBILDUNG 9-5: USER STORIES GENEHMIGEN, ABSCHÄTZEN UND ZUSAGEN—INPUTS, HILFSMITTEL UND OUTPUTS.....	204
ABBILDUNG 9-6: USER STORIES GENEHMIGEN, ABSCHÄTZEN UND ZUSAGEN—DATENFLUSSDIAGRAMM	205
ABBILDUNG 9-7: AUFGABEN ERSTELLEN—INPUTS, HILFSMITTEL UND OUTPUTS	210
ABBILDUNG 9-8: AUFGABEN ERSTELLEN—DATENFLUSSDIAGRAMM	210
ABBILDUNG 9-9: AUFGABENPLANUNGS-MEETINGS	ERROR! BOOKMARK NOT DEFINED.
ABBILDUNG 9-10: AUFGABEN ABSCHÄTZEN—INPUTS, HILFSMITTEL UND OUTPUTS	213
ABBILDUNG 9-11: AUFGABENSCHÄTZUNG—DATENFLUSSDIAGRAMM.....	214
ABBILDUNG 9-12: SPRINT BACKLOG ERSTELLEN—INPUTS, HILFSMITTEL UND OUTPUTS.....	222
ABBILDUNG 9-13: SPRINT BACKLOG ERSTELLEN—DATENFLUSSDIAGRAMM	223
ABBILDUNG 9-14: PLANUNGS- UND SCHÄTZUNGSPHASE—DATENFLUSSDIAGRAMM	227
ABBILDUNG 10-1: IMPLEMENTIERUNGSÜBERSICHT	230
ABBILDUNG 10-2: IMPLEMENTIERUNGSÜBERSICHT (KERNPUNKTE)	231
ABBILDUNG 10-3: ZU LIEFERNDE ERGEBNISSE ERZEUGEN—INPUTS, HILFSMITTEL UND OUTPUTS	232
ABBILDUNG 10-4: ZU LIEFERNDE ERGEBNISSE ERZEUGEN—DATENFLUSSDIAGRAM	233
ABBILDUNG 10-5: SCRUMBOARD	234
ABBILDUNG 10-6: TÄGLICHES STANDUP DURCHFÜHREN—INPUTS, HILFSMITTEL, AND OUTPUTS.....	239
ABBILDUNG 10-7: TÄGLICHES STANDUP DURCHFÜHREN—DATENFLUSSDIAGRAMM.....	240
ABBILDUNG 10-8: PRIORISIERTES PRODUKT BACKLOG PFLEGEN—INPUTS, HILFSMITTEL UND OUTPUTS	245
ABBILDUNG 10-9: PRIORISIERTES PRODUKT BACKLOG PFLEGEN—DATENFLUSSDIAGRAMM	246
ABBILDUNG 10-10: IMPLEMENTIERUNGSPHASE—DATENFLUSSDIAGRAMM	251
ABBILDUNG 11-1: ÜBERBLICK DER REVIEW- UND RÜCKBLICKSPHASE	255
ABBILDUNG 11-2: ÜBERBLICK DER REVIEW UND RÜCKBLICKSPHASE (KERNPUNKTE)	256
ABBILDUNG 11-3: ZUM SCRUM DER SCRUMS ZUSAMMENKOMMEN—INPUTS, HILFSMITTEL UND OUTPUTS... ERROR! BOOKMARK NOT DEFINED.	
ABBILDUNG 11-4: ZUM SCRUM DER SCRUMS ZUSAMMENKOMMEN—DATENFLUSSDIAGRAMM	ERROR! BOOKMARK NOT DEFINED.
ABBILDUNG 11-5: SPRINT VORFÜHREN UND VALIDIEREN—INPUTS, HILFSMITTEL UND OUTPUTS	257
ABBILDUNG 11-6: SPRINT VORFÜHREN UND VALIDIEREN—DATENFLUSSDIAGRAMM	258

ABBILDUNG 11-7: SPRINT RÜCKBLICK HALTEN—INPUTS, HILFSMITTEL UND OUTPUTS.....	262
ABBILDUNG 11-8: SPRINT RÜCKBLICK HALTEN— DATENFLUSSDIAGRAMM.....	262
ABBILDUNG 11-9: ÜBERSICHT UND RÜCKBLICKSPHASE—DATENFLUSSDIAGRAMM	268
ABBILDUNG 12-1: RELEASE ÜBERSICHT	270
ABBILDUNG 12-2: RELEASE ÜBERSICHT (KERNPUNKTE)	271
ABBILDUNG 12-3: ZU LIEFERNDE ERGEBNISSE AUSLIEFERN—INPUTS HILFSMITTEL UND OUTPUTS.....	272
ABBILDUNG 12-4: ZU LIEFERNDE ERGEBNISSE AUSLIEFERN – DATENFLUSSDIAGRAMM	273
ABBILDUNG 12-5: PROJEKT RÜCKBLICK HALTEN—INPUTS, HILFSMITTEL UND OUTPUTS.....	277
ABBILDUNG 12-6: PROJEKT RÜCKBLICK HALTEN—DATENFLUSSDIAGRAMM.....	277
ABBILDUNG 12-7: RELEASEPHASE—DATENFLUSSDIAGRAMM.....	281

Tabellenverzeichnis

TABELLE 1-1: ÜBERSICHT ÜBER SCRUM PROZESSE	15
TABELLE 1-2: SCRUM VERSUS TRADITIONELLES PROJEKTMANAGEMENT	ERROR! BOOKMARK NOT DEFINED.
TABELLE 3-1: VERANTWORTLICHKEITEN DES PRODUCT OWNERS IN SCRUM PROZESSEN	ERROR! BOOKMARK NOT DEFINED.
TABELLE 3-2: VERANTWORTLICHKEITEN DES SCRUM MASTERS IN SCRUM PROZESSEN	ERROR! BOOKMARK NOT DEFINED.
TABELLE 3-3: VERANTWORTLICHKEITEN DES SCRUM TEAMS IN DEN SCRUM PROZESSEN	52
TABELLE 3-4: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN IN BEZUG ZUR ORGANISATION	57
TABELLE 4-1: ERTRAGSWERT FORMELN	79
TABELLE 4-2: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN IN BEZUG ZUR GESCHÄFTLICHEN RECHTFERTIGUNG	83
TABELLE 5-1: STUFEN DER ABNAHMEKRITERIEN	ERROR! BOOKMARK NOT DEFINED.
TABELLE 5-2: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH QUALITÄT	98
TABELLE 6-1: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN HINSICHTLICH ÄNDERUNGEN	121
TABELLE 7-1: ZUSAMMENFASSUNG DER VERANTWORTLICHKEITEN IM BEZUG AUF RISIKEN	138

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 Stakeholder 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. 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.

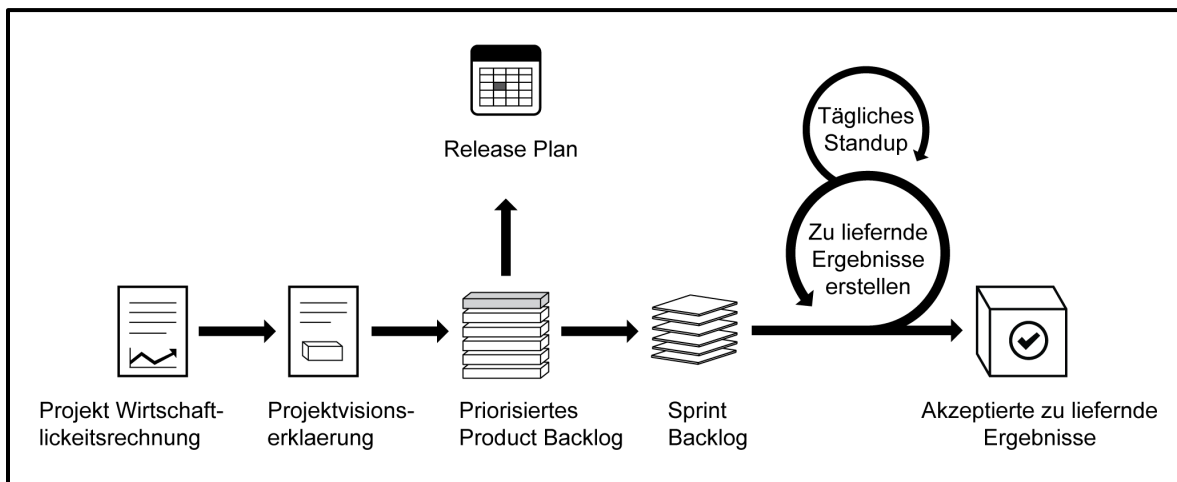


Abbildung 1-1: Scrum Flussdiagramm

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

Auf der OOPLSA (Objektorientiertes Programmieren, Systeme, Sprachen und Applikationen) Konferenz im Jahr 1995 in Austin, Texas haben Ken Schwaber und Jeff Sutherland das Scrum Konzept und seine Anwendbarkeit auf die Softwareentwicklung in Präsentationen näher ausgeführt. Seit diesem Zeitpunkt haben mehrere Scrum Praktiker, Experten und Autoren die Scrum Konzeptualisierung und Methodik kontinuierlich verfeinert. In den letzten Jahren hat Scrum an Popularität zugenommen und ist jetzt die präferierte Projektentwicklungsmethode für viele Organisationen weltweit.

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 vorführen und validieren* gegeben.
4. **Kontinuierliche Verbesserung** — Durch den Prozess *Priorisiertes Produkt Backlog pflegen* 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 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 Zusage* 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 *Projekt 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™

In den letzten Jahren hat es sich gezeigt, dass Organisationen, die Scrum als ihren bevorzugten Rahmen für Projekte anwenden, sehr hohe Kapitalerträge erwirtschaften. 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. Stakeholder-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 Stakeholder (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 und 14 beinhalten die zusätzlichen Prozesse, die spezifisch bei der Skalierung von Scrum

Abbildung 1-2 zeigt die SBOK™ Rahmenstruktur, welche erklärt, dass Prinzipien, Aspekte und Prozesse miteinander interagieren und von gleicher Bedeutung für ein besseres Verständnis der Scrum Rahmenstruktur sind.

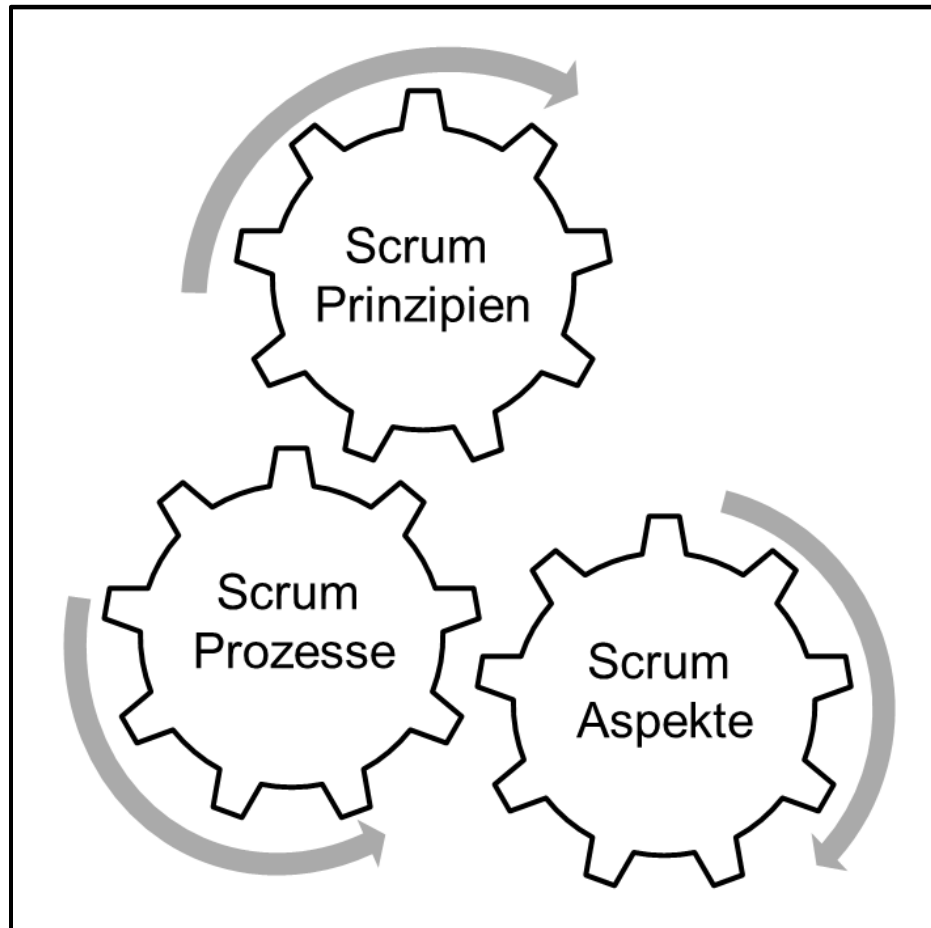


Abbildung 1-2: SBOK™ Rahmenstruktur

1.4.1 Wie wende ich den SBOK™ an?

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 fuer 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 Praktiken 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 basieren auf dem spezifischen Bedarf der jeweiligen Organisation, des Projekts, Produkts oder Teams angewendet. Zusätzliche Prozesse kommen zur Anwendung, wenn Scrum für große Projekte skaliert wird (Kapitel 13.) oder bei der Skalierung für das Unternehmen (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.

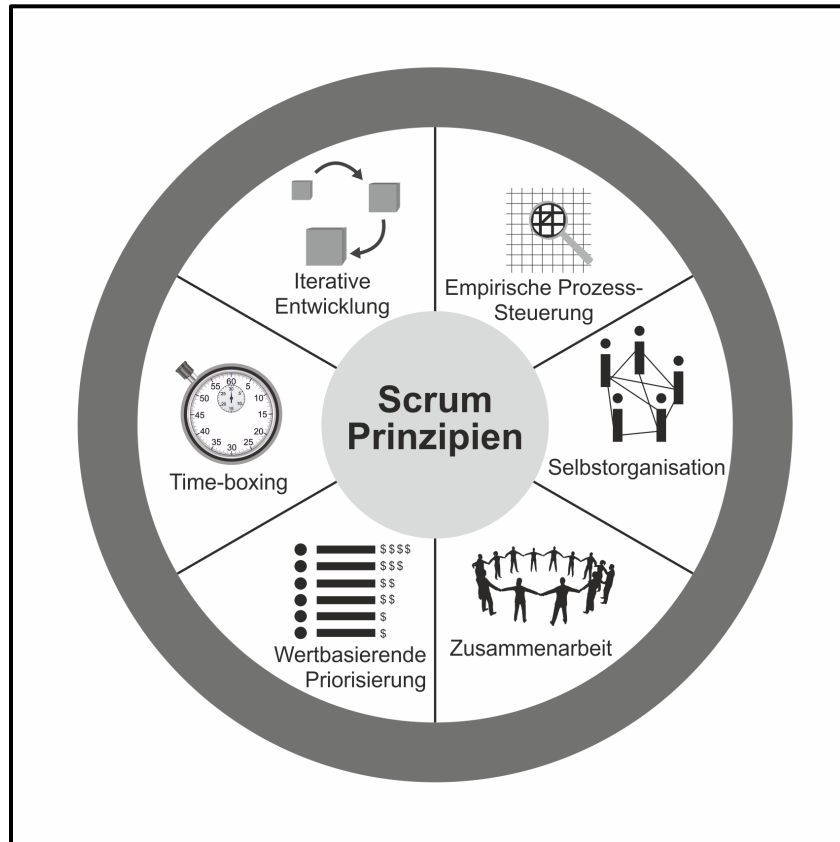


Abbildung 1-3: Scrum Prinzipien

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.
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 zusammenarbeiten und miteinander 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:

- **Der Stakeholder**, 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 Stakeholder, für die durch das Projekt ein kollaborativer Nutzen hergestellt wird.
- **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 eine Scrum Organisationsstruktur.

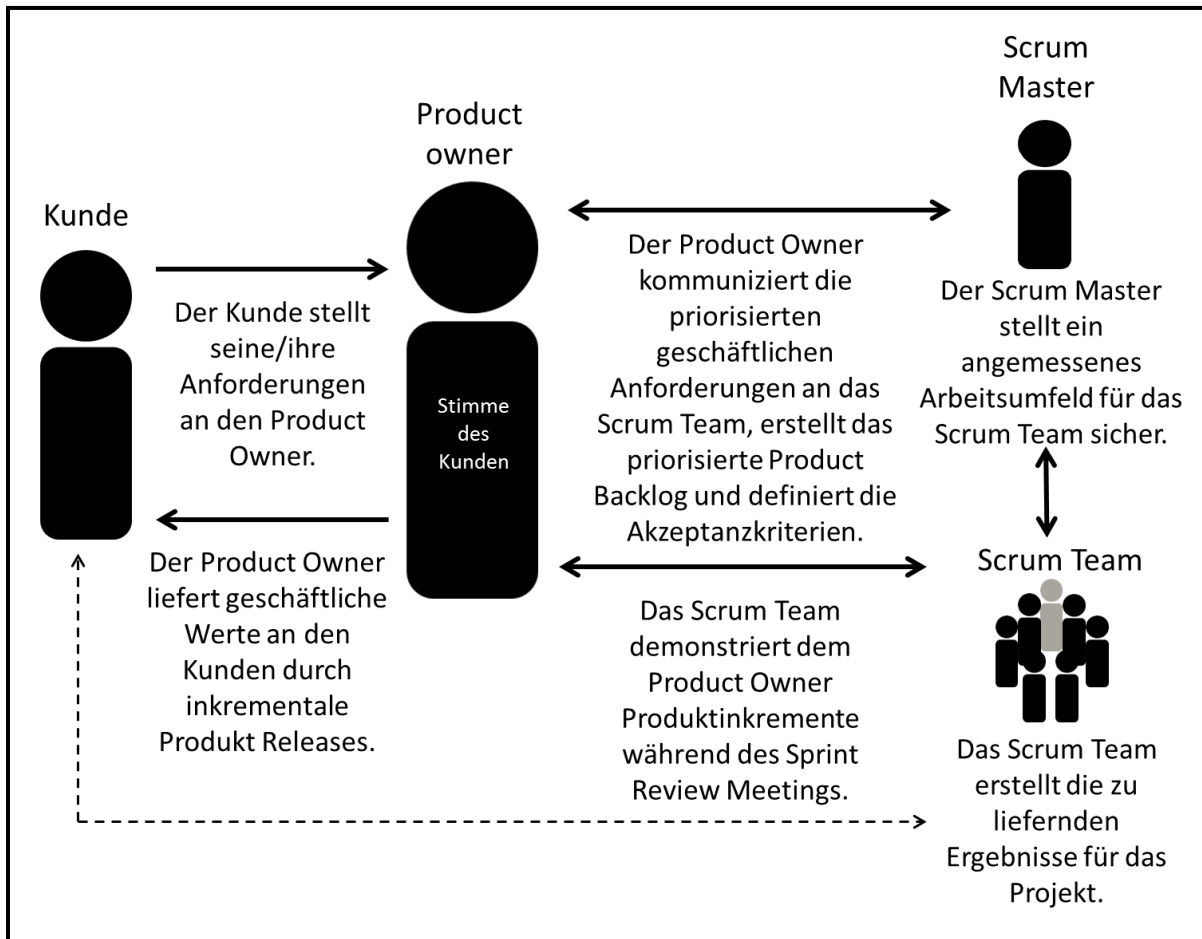


Abbildung 1-4: Organisation in Scrum

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 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 Stakeholder 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 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) Stakeholder (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 Stakeholder 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. 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 Stakeholder 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 erstellen
10	Implementierung	13. Zu liefernde Ergebnisse erzeugen 14. Tägliche Standups durchführen 15. Priorisiertes Produkt Backlog pflegen
11	Rückblick und Retrospektion	16. Sprint durchführen und validieren 17. Sprint Rückblick halten
12	Release	18. Zu liefernde Ergebnisse ausliefern 19. Projekt 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, gibt es drei zusätzliche Scrum Prozesse, die in Kapitel 13 - Die Skalierung von Scrum für große Projekte - definiert sind. Es gibt auch spezifisch Prozesse, die die Anwendung von Scrum auf der Ebene des Unternehmens

definieren. Diese sind in Kapitel 14 enthalten - Die Skalierung von Scrum für das Unternehmen. Diese zusätzlichen Scrum Prozesse sind in Tabelle 1-2 zusammengefasst.

Kapitel	Anwendbarkeit	Zusätzliche Scrum-Prozesse
13	Scrum für große Projekte Skalieren	<ol style="list-style-type: none"> 1. Komponenten des Großen Projekts Erstellen 2. Sprints Durchführen und Koordinieren 3. Ein Release im Großen Projekt Vorbereiten
14	Scrum für das Unternehmen Skalieren	<ol style="list-style-type: none"> 1. Programm- oder Portfolio- Komponenten Erstellen 2. Scrum Guidance Body Reviewen und Aktualisieren 3. Programm- oder Portfolio- Backlog Erstellen und Pflegen 4. Programm- oder Portfolio- Komponenten Koordinieren 5. Programm oder Portfolio Releases Rückblick Halten

Tabelle 1-2: zusammengefasst

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 Stakeholder identifizieren* — In diesem Prozess werden der Scrum Master und der/die Stakeholder 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 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 Stakeholder 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 erstellen* — 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 pflegen* — 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 **Rückblick und Retrospektion**

16. *Sprint vorführen und validieren* — In diesem Prozess führt das Scrum Team die zu liefernden Ergebnisse des Sprints dem Product Owner und den relevanten Stakeholdern in einem Sprint Rückblick Meeting vor. Der Zweck dieses Meetings ist es, sich die Zustimmung und Akzeptanz für die zu liefernden Ergebnisse, die in dem Sprint hergestellt wurden, vom Product Owner 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 Stakeholder weitergeleitet. Eine formale Vereinbarung über funktionierende zu liefernde Ergebnisse dokumentiert den erfolgreichen Abschluss des Sprints.
19. *Projekt Rückblick halten* — In diesem Prozess, der das Projekt abschließt, kommen organisatorische Stakeholder 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 für große Projekte Skalieren

20. Komponenten des Großen Projekts Erstellen — dieser Prozess definiert wie die verschiedenen Product Owner als Team zusammenarbeiten, und wie die verschiedenen Scrum Teams zusammenarbeiten. Außerdem werden gemeinsame Komponenten Bestandteile sowie gemeinsame und spezialisierte Ressourcen identifiziert.
21. Sprints Durchführen und Koordinieren — dieser Prozess ist nur für große Projekte relevant und spricht bestimmte Aspekte an, die während jedes Sprints in Erwägung gezogen werden sollten. Falls erforderlich, werden Scrum der Scrums Meetings durchgeführt, um die Arbeiten zwischen verschiedenen Scrum Teams zu koordinieren.
22. Ein Release im Großen Projekt Vorbereiten — In einigen großen Projekten kann es geschäftlich Sinn machen, vor einem Release einen speziellen Sprint zur Vorbereitung dieses Releases durchzuführen (das wird vom Projekt Team entsprechend den geschäftlichen Notwendigkeiten entschieden). Dieser Prozess behandelt einen solchen Vorbereitungs-Sprint. Die Existenz eines Vorbereitungs-Sprints darf nicht dazu führen, dass Aktivitäten, die in die anderen Sprints gehören, bis zu diesem Punkt verschoben werden.

1.4.4.7 Scrum für das Unternehmen Skalieren

23. Programm- oder Portfolio- Komponenten Erstellen — in diesem Prozess identifizieren Programm- oder Portfolio- Product Owner und relevante Stakeholder gemeinsame Komponenten und Ressourcen, die für das Programm oder Portfolio erforderlich sind. Die minimalen Done Kriterien werden definiert, und alle übrigen Stakeholder werden identifiziert.
24. Scrum Guidance Body Reviewen und Aktualisieren — in diesem Prozess werden die Empfehlungen des Scrum Guidance Body regelmäßig von den Mitgliedern des Scrum Guidance Bodys überprüft und, falls notwendig, aktualisiert. In diesem Prozess werden auch Änderungen in der Mitgliedschaft des Scrum Guidance Bodys gehandhabt.
25. Programm- oder Portfolio- Backlog Erstellen und Pflegen — in diesem Prozess wird das Programm oder Portfolio Backlog erstellt, aktualisiert und gepflegt. Empfehlungen für Verbesserungen der Empfehlungen des Scrum Guidance Body können gemacht werden. Dem Fortschritt entsprechend und/oder auf Grundlage von geänderten Anforderungen der Projekte des Programms oder Portfolios werden die Zieltermine für die Implementierung gegebenenfalls angepasst.
26. Programm- oder Portfolio- Komponenten Koordinieren — in diesem Prozess werden die Komponenten des Programms oder Portfolios koordiniert. Abhängigkeiten zwischen Projekten werden adressiert, gemeinsame Hindernisse werden erörtert, und Beste Praktiken werden ausgetauscht. Manchmal werden Empfehlungen für Verbesserungen des Scrum Guidance Bodys gemacht.

27. Programm oder Portfolio Releases Rückblick Halten — in diesem Prozess kommen der Programm oder Portfolio Product Owner und relevante Stakeholder zusammen, um auf ein Programm- oder Portfolio- Release zurückzublicken und Lehren aus dem Release zu identifizieren, zu dokumentieren und zu verinnerlichen. Oft führen diese Lehren zur Dokumentation von vereinbarten Maßnahmen, die in zukünftigen Releases umgesetzt werden. Manchmal werden Empfehlungen zur Verbesserung des Scrum Guidance Body ausgesprochen.

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 1-3: 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 Stakeholder 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 Werterzeugender Prozess ist, bei welchemdie Zusammenarbeit aller Stakeholder 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 und das Sprint-Review 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 Stakeholder 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 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 Team, eingesehen werden kann
- Ein Release Plan, welcher wenn nötig auch über mehrere Scrum Teams hinweg koordiniert wird
- Klare Einsicht in die Fortschritte des Teams anhand der Verwendung eines Scrumboards, eines Burndown-Charts, und anderer Hilfsmittel zur Informationsverbreitung
- Bei der *Durchführung der täglichen Standup Meetings* berichten alle Teammitglieder, 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.
- Bei den Sprint Review Meetings, welche während Sprint vorführen und validieren durchgeführt werden, gibt das Scrum Team Einblick in potentiell versandfähige zu liefernde Ergebnisse.

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

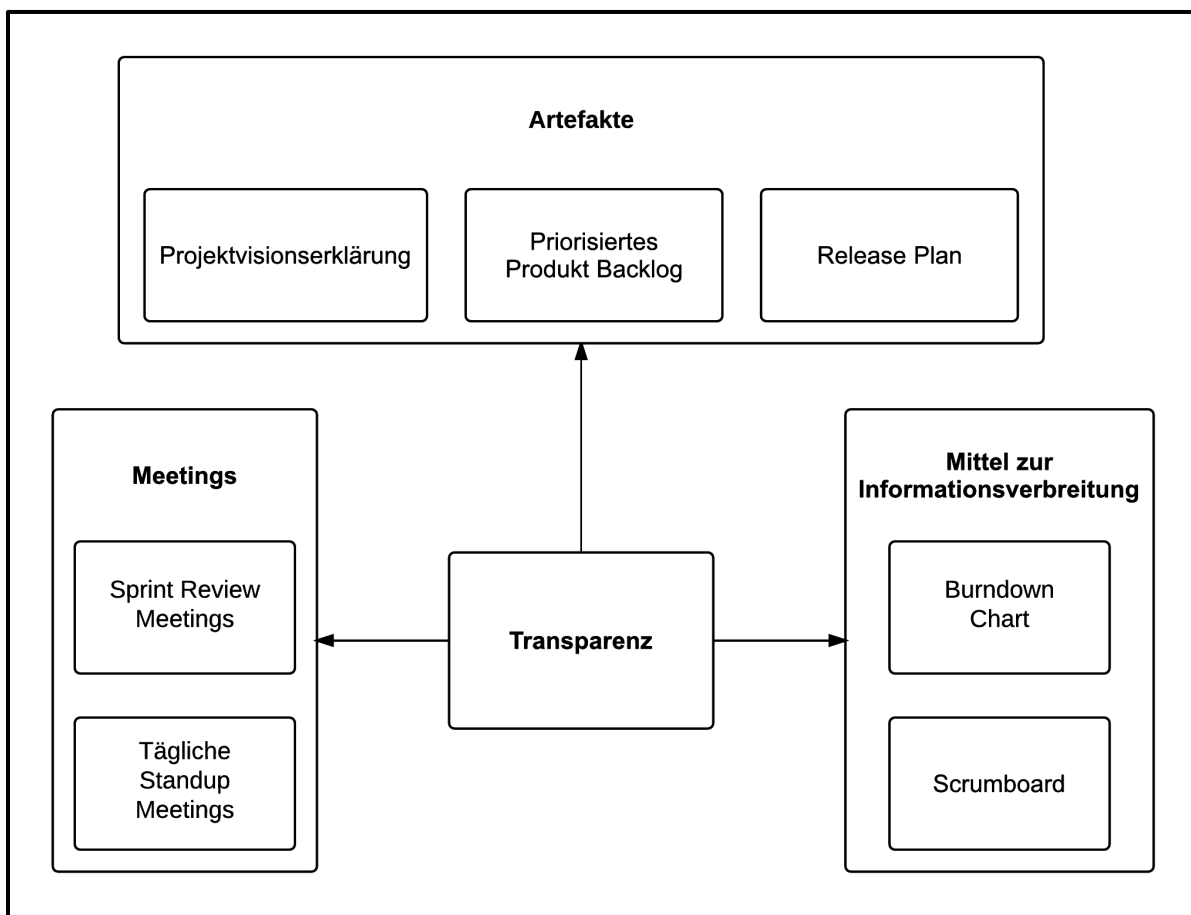


Abbildung 2-1: Transparenz bei Scrum

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 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.

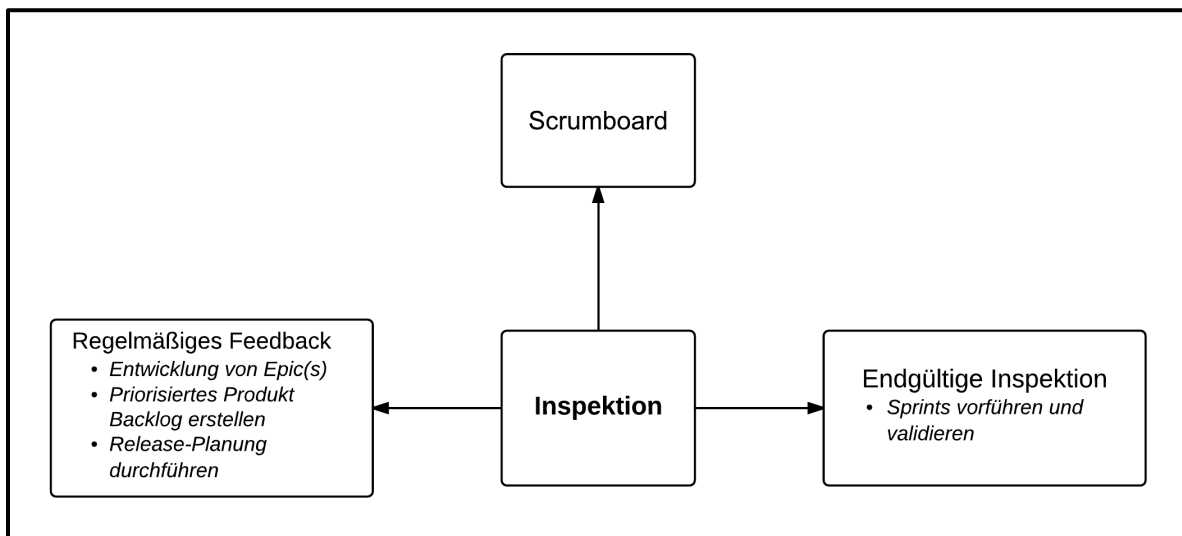


Abbildung 2-2: Inspektion bei Scrum

2.3.3 Adaptierung

Adaptierung geschieht, wenn das Scrum Kernteam und die Stakeholder durch Transparenz und Inspektion lernen und dann ihre Arbeitsweise an die erlangten Erkenntnisse anpassen. Einige Beispiele für Adaptierung 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 pflegen*, und *Sprint vorführen und validieren*.
- Verbesserungen können auch zu Change Requests führen, welche während *Epic(s) entwickeln*, *Priorisiertes Produkt Backlog erstellen*, und *Priorisiertes Produkt Backlog pflegen* 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 pflegen*, um Anleitung und Expertise zu bieten, so diese benötigt wird.
- Bei *Sprint-Rückblick durchführen* werden vereinbarte Verbesserungsmaßnahmen aufgrund der Ergebnisse aus *Sprint vorführen und validieren* festgelegt.
- Beim Projektrückblick-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.

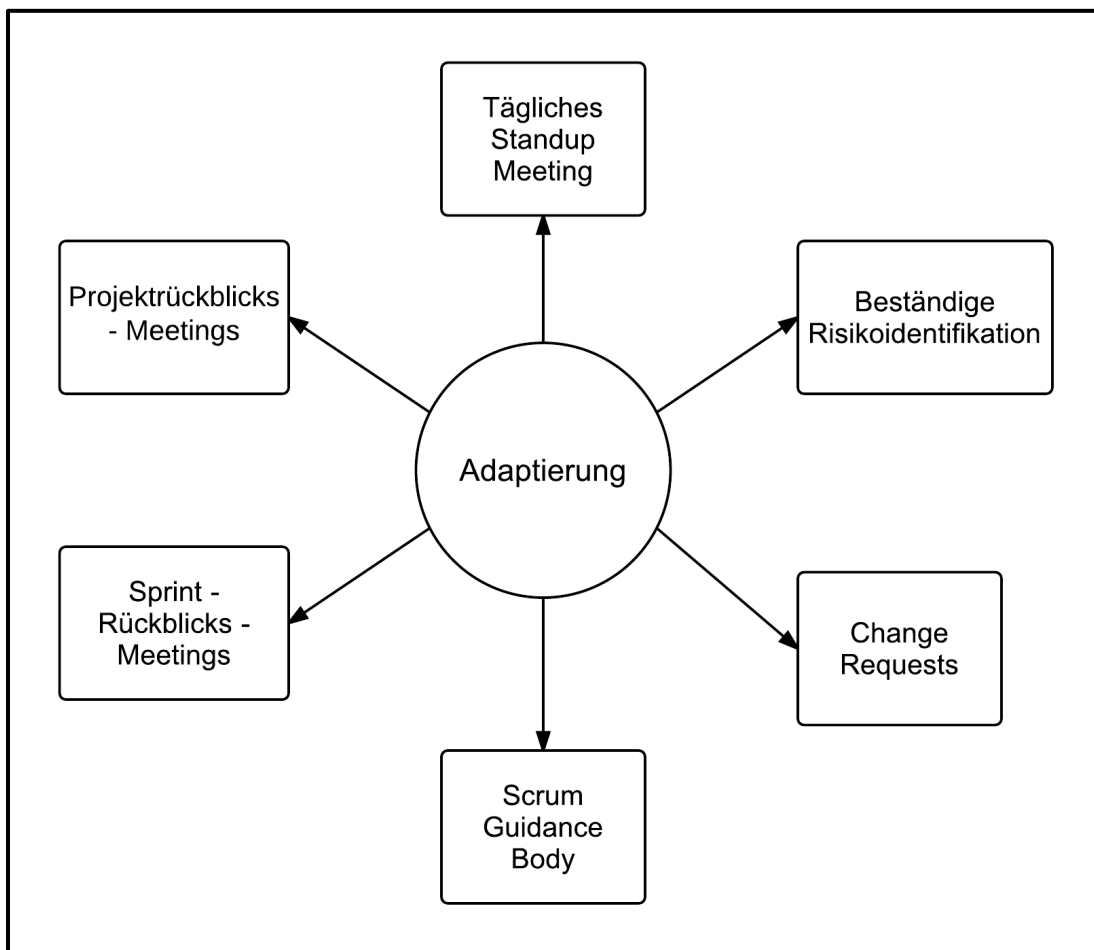


Abbildung 2-3: Adaptierung in Scrum

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.

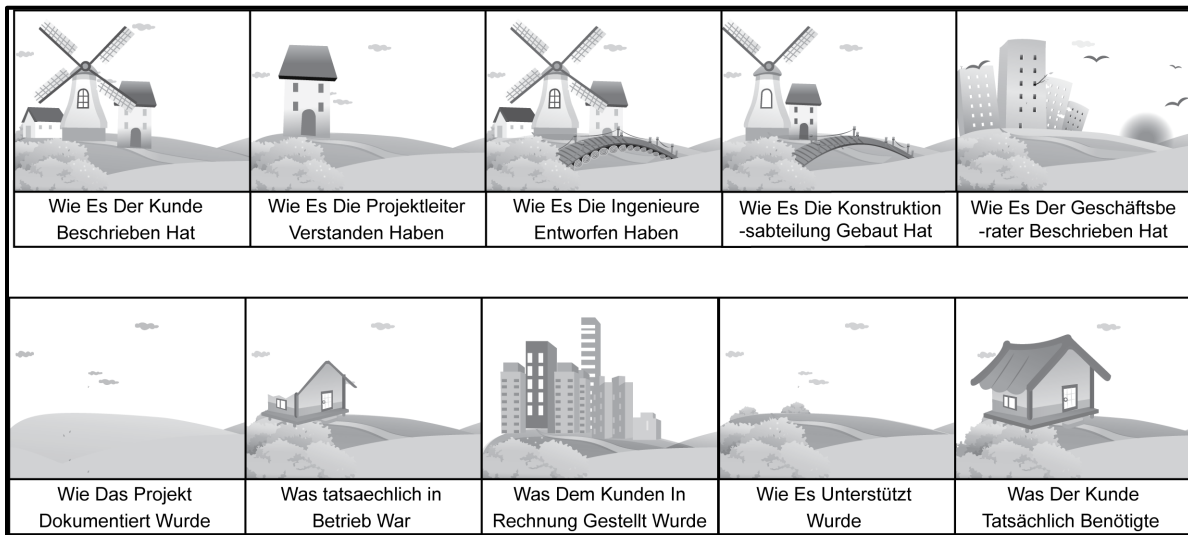


Abbildung 2-4: Probleme beim Wasserfall-Projektmanagement

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 der "Dienende Leader", 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.

2

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.

Selbstorganisation bedeutet nicht, dass die Teammitglieder dazu angehalten sind, sich so zu verhalten, wie es ihnen gerade in den Sinn kommt. Es bedeutet ganz einfach, dass, sobald die *Erstellung der Produktvision* abgeschlossen ist, der Product Owner, Scrum Master und das Scrum Team identifiziert werden, und dass das Scrum Core Team selbst sehr eng mit den relevanten Stakeholdern zur besseren Verfeinerung der Anforderungen während *Epic(s) entwickeln* und *User Storys erstellen* zusammenarbeiten. 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. 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 gefertigte zu liefernde Ergebnisse akzeptiert werden, eng zusammen, um den Teil des Projektes, welcher im Sprint erzeugt wurde, vorzustellen. Da zu liefernde Ergebnisse potentiell 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 Stakeholdern akzeptiert wird.

Die Hauptziele von selbstorganisierenden Teams lauten wie folgt:

- 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 Erstellen*.
- 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 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.

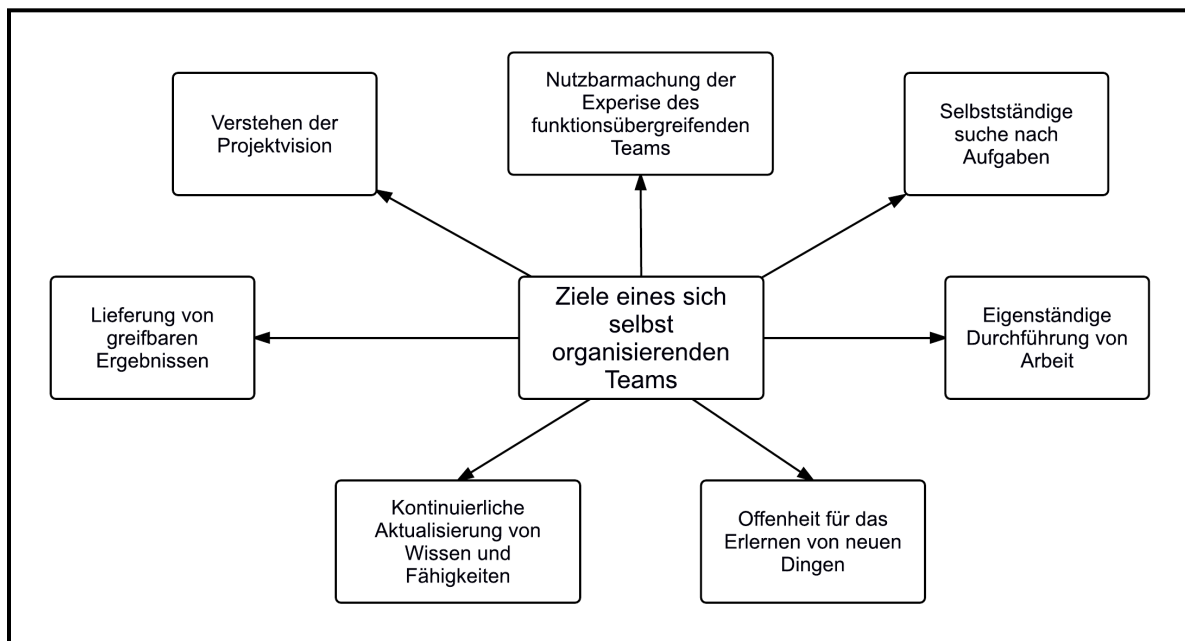


Abbildung 2-5: Ziele eines sich selbst organisierenden Teams

2.5 Kollaboration

Kollaboration bezieht sich im Rahmen vom Scrum auf die Zusammenarbeit des Scrum Kernteams und dessen Absprache mit den 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.

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 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 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 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 pflegen* 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.

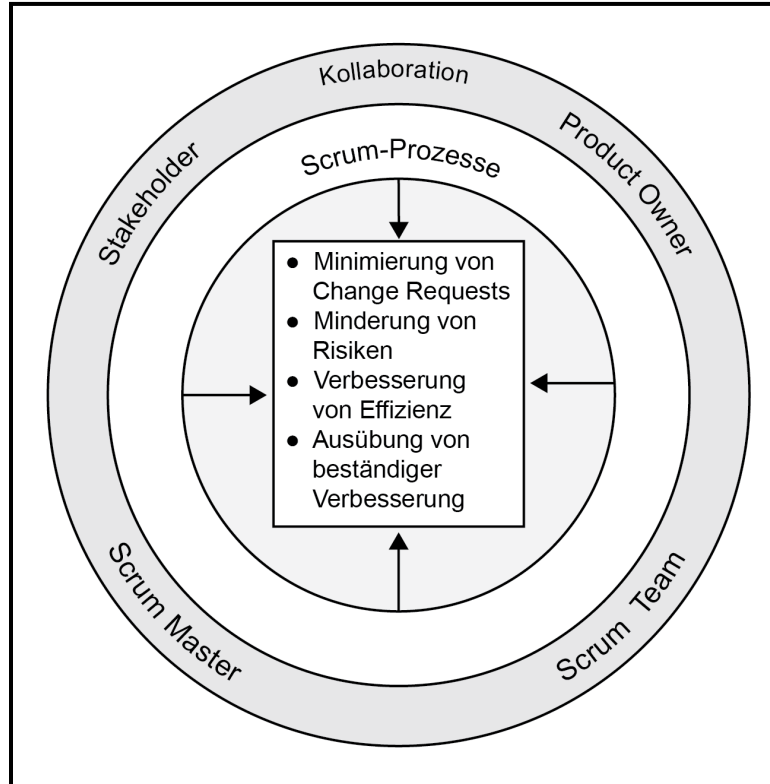


Abbildung 2-6: Vorteile der Kollaboration bei einem Scrum-Projekt

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.

Kollaborationsmittel, welche für ko-lokalisierte oder verteilte Teams angewandt werden können sind:

1. **Co-lokalisierte Teams** (Teams, die im selben Büro arbeiten) — Bei Scrum werden ko-lokalisierte Teams bevorzugt. In diesem Fall sind die bevorzugten Methoden der Kommunikation unter anderem Interaktionen von Angesicht zu Angesicht, Räume für Entscheidungen und War Rooms, Scrumboards, Wandtafeln, gemeinsame Tische etc.
2. **Verteilte Teams** (Teams, deren Mitglieder von unterschiedlichen physischen Orten aus arbeiten) — Obwohl ko-lokalisierte Teams bevorzugt werden, können auf Grund von Outsourcing, Offshoring, unterschiedlichen Standorten, Heimarbeitsoptionen etc. Scrum Teams mitunter verteilt sein. Einige Hilfsmittel welche für effektive Kollaboration bei verteilten Teams verwendet werden könnten, sind unter anderem Videokonferenzen, Instant-Messengerdienste, Chats, Soziale Medien, Gemeinsame Bildschirme, und Software zur Simulation der Funktion von Scrumboards, Wandtafeln usw.

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, und diese in Form von bearbeitbaren User Storys niedergeschrieben hat, arbeitet sie/er mit dem Kunden und dem Sponsor zusammen, um zu verstehen, welche geschäftlichen Anforderungen den maximalen Geschäftswert bieten. 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. Mitunter kann es sein, das ein Kunde alle User Storys als höchste Priorität festlegt. Während dies durchaus zutreffen kann, muss selbst dann die Liste der User Storys mit höchster Priorität innerhalb der Liste priorisiert werden. Die Priorisierung eines Backlogs umfasst die kritische Betrachtung jeder User Story. 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 pflegen*.

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 Stakeholder 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.

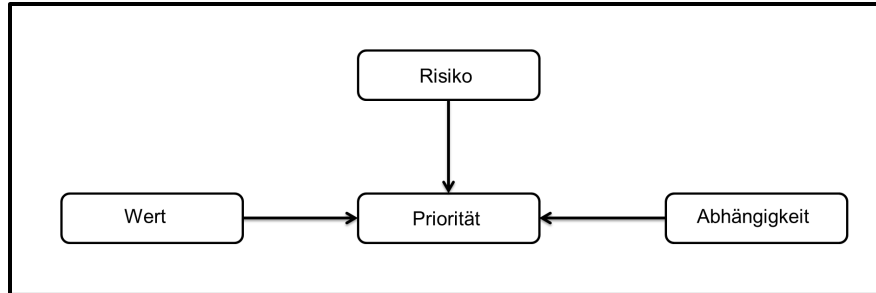


Abbildung 2-7: Wertbasierte Priorisierung

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

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. Arbitrarisches 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 versandfähige Produkt-Funktionalitäten umzuwandeln. Um die größten Vorteile aus einem Scrum Projekt zu ziehen, ist es immer ratsam, den Sprint auf maximal 4 Wochen zu begrenzen, außer bei Projekten mit sehr stabilen Anforderungen, bei welchen die Sprints auf bis zu 6 Wochen ausgedehnt werden können.
- **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 Tägliches Standup Meetings durchführen.

- **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 erstellen durchgeführt. Es ist auf 8 Stunden für einen einmonatigen Sprint begrenzt. Das Sprint-Planungs Meeting ist in zwei Teile geteilt:
 - Zieldefinition — 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.
- **Sprint-Review Meeting** — Das Sprint Review Meeting ist auf vier Stunden bei einem einmonatigen Sprint begrenzt. 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 4 Stunden bei einem einmonatigen Sprint begrenzt und 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 Time-Boxing-Länge bei Scrum Meetings.

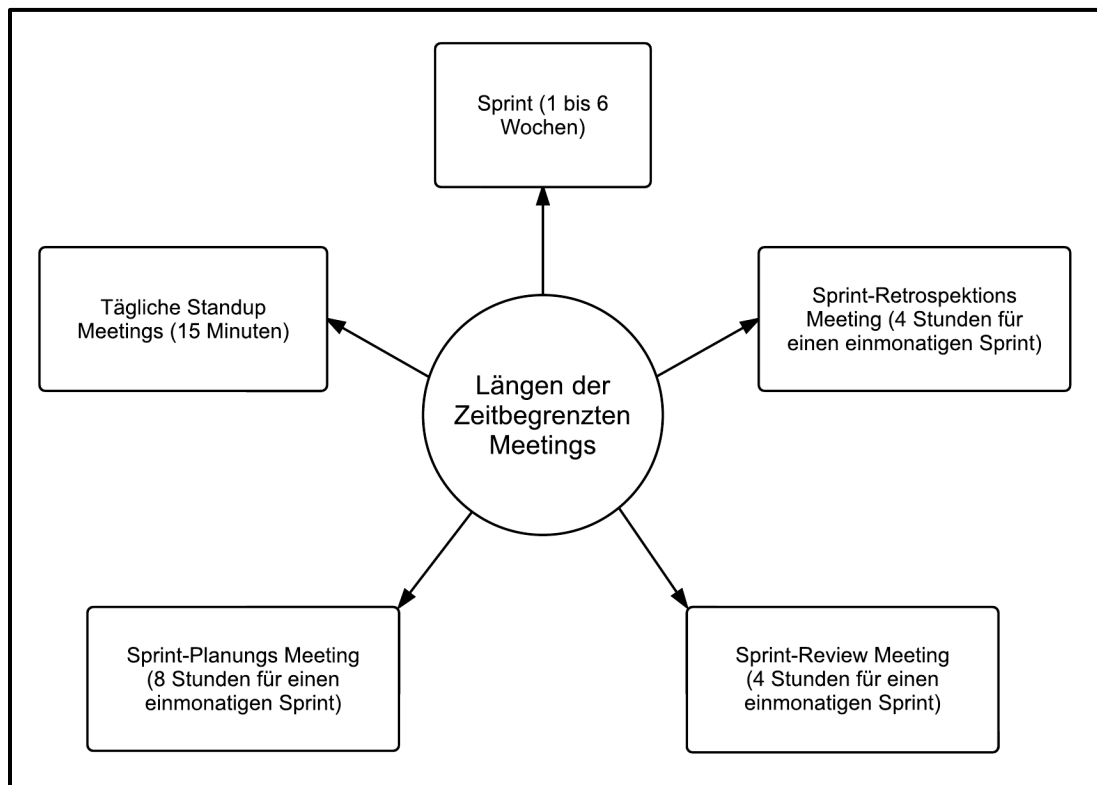


Abbildung 2-8: Time-Boxing-Situationen bei Scrum Meetings

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 pflegen* 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 erstellen* 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 potentiell 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.

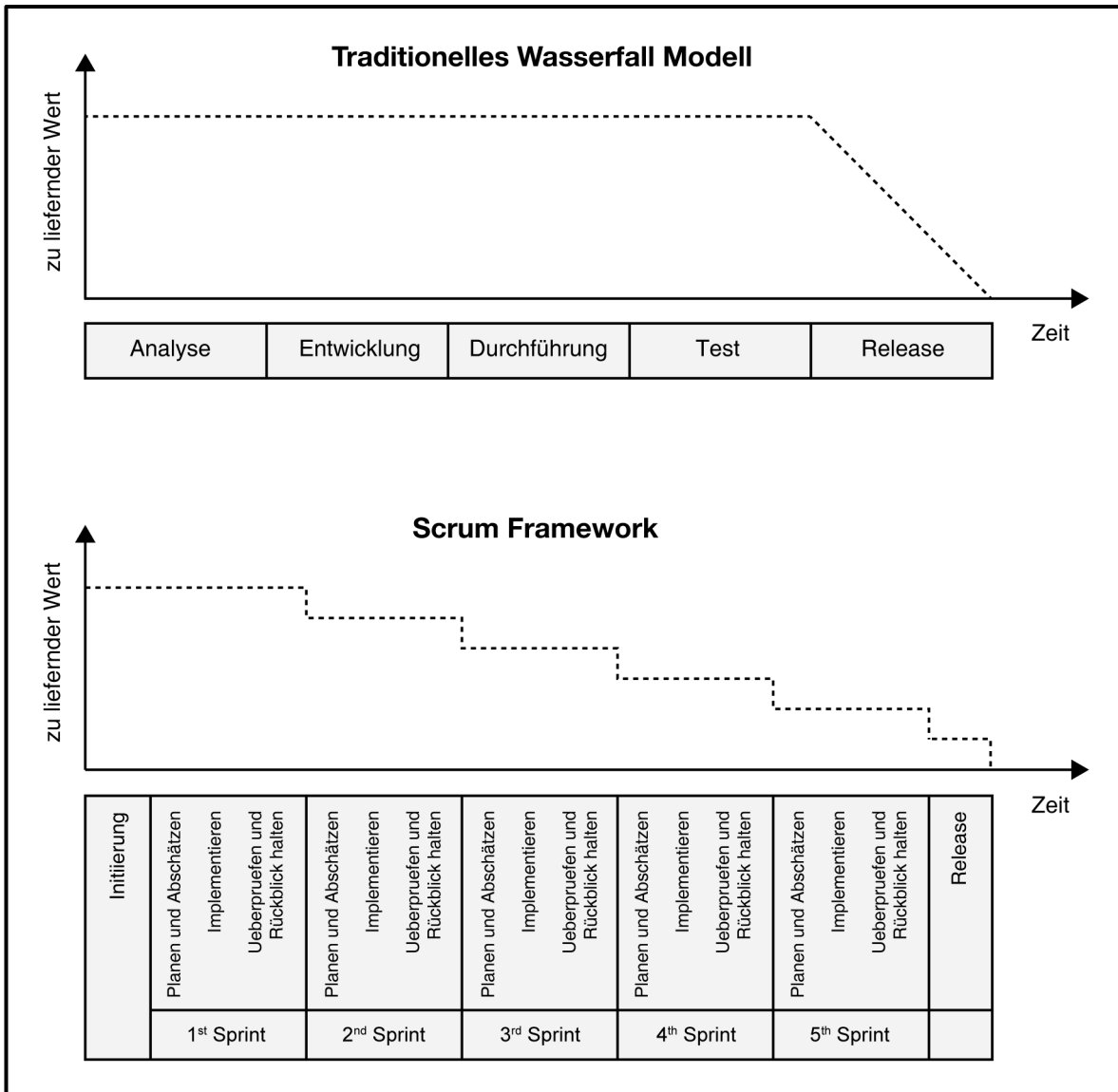


Abbildung 2-9: Scrum vs. Traditionelles Wasserfall Modell

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, dass 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 Stakeholder 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.

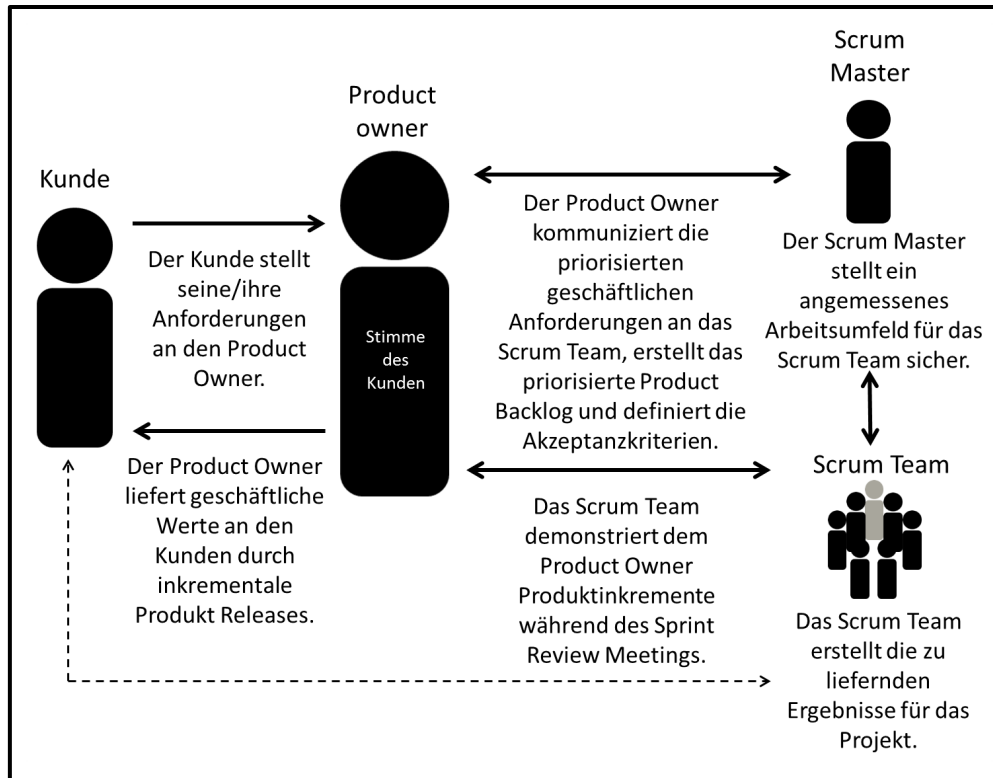


Abbildung 3-1: Scrum Rollen—Überblick

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. Stakeholder

Stakeholder 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. Stakeholder beeinflussen das Projekt die ganze Projektentwicklung hindurch. Stakeholder können auch während *Epics entwickeln*, *Priorisiertes Produkt Backlog erstellen*, *Release-Planung durchführen*, *Sprint Rückblick halten* und *anderen wichtigen Prozessen* in Scrum eine Rolle spielen.

- **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 Stakeholder dem letztendlich alle Rechenschaft ablegen.

Manchmal kann dieselbe Person oder Organisation mehrere Stakeholder Rollen übernehmen; zum Beispiel können der Sponsor und Kunde derselbe sein.

2. Anbieter

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

3. 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 Stakeholder 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 Befürfnisse aller Stakeholder verstehen und unterstützen, gleichzeitig aber auch die Bedürfnisse und Tätigkeiten des Scrum Teams verstehen. Da der Product Owner die Bedürfnisse und Prioritäten der Stakeholder, 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 Identifizierung von Scrum Master und Stakeholder(n)	<ul style="list-style-type: none"> • Hilft den Scrum Master für das Projekt festzulegen • Identifiziert Stakeholder
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
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"> • Hilft User Storys zu erstellen • 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 erstellen	<ul style="list-style-type: none"> • Hilft dem Scrum Team das Sprint Backlog zu erstellen
10.1 Zu liefernde Ergebnisse erzeugen	<ul style="list-style-type: none"> • Verdeutlicht dem Scrum Team Geschäftliche Anforderungen
10.3 Priorisiertes Produkt	<ul style="list-style-type: none"> • Pflegt das Priorisierte Produkt Backlog

Backlog pflegen	
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 das Produkt Release zu verteilen und koordiniert dies mit dem Kunden
12.2 Projekt Rückblick halten	<ul style="list-style-type: none"> • Nimmt an Sprint 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

3.4.1 Stimme des Kunden (VOC)

Als Repräsentant der Stakeholder 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.

3.4.2 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 Masters 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.

3.4.3 Program Product Owner

Er/sie definiert die strategischen Ziele und Prioritäten für das Programm. Der Programm Product Owner ist dafür verantwortlich, den Geschäftswert eines Programms zu maximieren. Er oder sie ist dafür verantwortlich, Kundenanforderungen zu artikulieren und die geschäftliche Rechtfertigung für das Programm zu aktualisieren, und kann wertvolle Impulse für die Visionen der Projekte in einem Programm geben. Der Programm Product Owner verwaltet auch das Programm Produkt Backlog.

Der Programm Product Owner spricht sich mit dem Portfolio Product Owner ab, um sicherzustellen, dass das Programm mit den Zielen des Portfolios kompatibel ist. Er oder sie ist auch daran beteiligt, Product Owner für einzelne Projekte zu ernennen, und sicherzustellen, dass die Visionen, Ziele, Ergebnisse, und Releases von einzelnen Projekten mit denen des Programms in Einklang stehen.

Diese Rolle ist der des Product Owners ähnlich, mit dem Unterschied, dass sie den Bedarf des Programms- oder des Geschäftsbereichs statt eines einzelnen Scrum Teams abdeckt.

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.

3.4.4 Portfolio Product Owner

Der Portfolio Product Owner trifft die Entscheidungen auf Portfolioebene. Er/sie hat die beste Perspektive, um zu entscheiden, wie das Unternehmen zu organisieren ist, um der Vision zu entsprechen. Die Rolle des

Portfolio Product Owners ist der des Programm Product Owners in einem Programm sehr ähnlich. Er/sie ist verantwortlich für die Schaffung und die Pflege des Portfolio Produkt Backlogs und treibt diese voran.

Diese Rolle ist der des Product Owners ähnlich, mit dem Unterschied, dass sie den Bedarf des Programms- oder des Geschäftsbereichs statt eines einzelnen Scrum Teams abdeckt.

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.

3.5 Scrum Master

Der Scrum Master ist der „Dienende 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 Identifizierung von Scrum Master und Stakeholder(n)	<ul style="list-style-type: none"> Hilft den/die Stakeholder 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 und der Definierung der Done-Kriterien
8.6 Release-Planung durchführen	<ul style="list-style-type: none"> Koordiniert die Erstellung des Release Plans Bestimmt die Sprint-Länge (mit Team und Product Owner)
9.1 User Storys Erstellen	<ul style="list-style-type: none"> Unterstützt das Scrum Team in der 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 erstellen	<ul style="list-style-type: none"> Unterstützt das Scrum Team der Erstellung des Sprint Backlogs und 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 pflegen	<ul style="list-style-type: none"> Unterstützt Priorisiertes Produkt Backlog Review Meetings
11.1 Zum Scrum der Scrum 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 Projekt 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

3.5.1 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 3-2 bietet Fragen die während eines Scrum der Scrums Meeting gestellt werden.

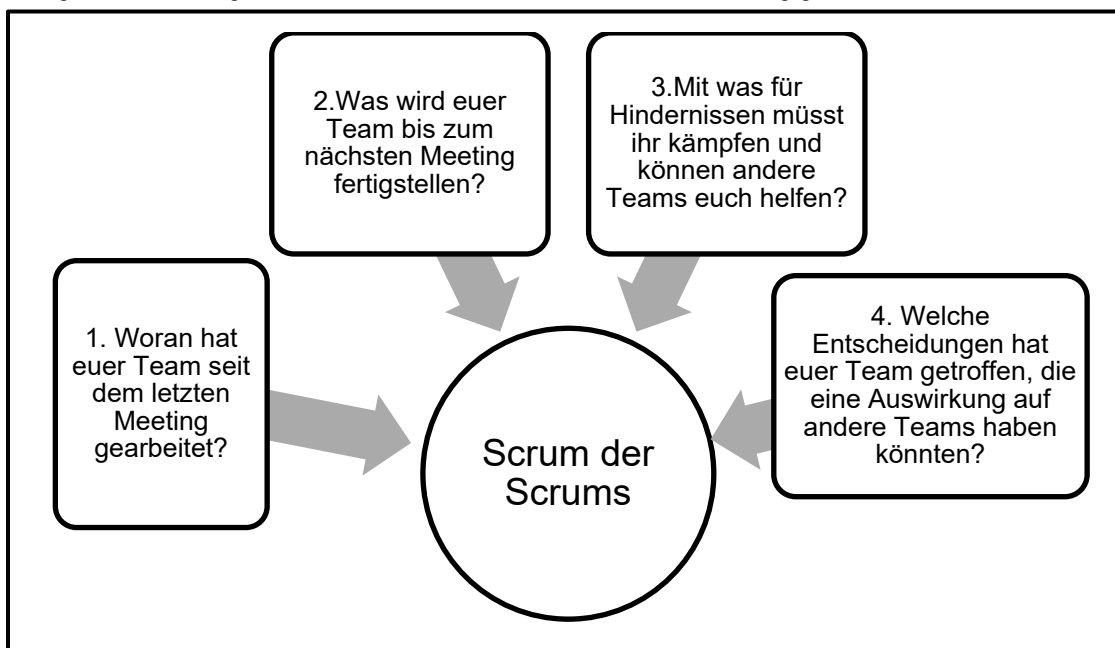


Abbildung 3-2: Gestellte Fragen während eines Scrum der Scrums Meeting

Der Chief Scrum Master kann aus der Gruppe der Scrum Master des großen Projekts ausgewählt werden, er kann aber auch jemand anderes sein. Für sehr große Projekte wird empfohlen, dass der Chief Scrum Master nicht auch gleichzeitig ein Scrum Master ist, da der erforderliche Aufwand seiner Rolle es dem Chief Scrum Master nicht erlauben wird, seinem Scrum Team genug Zeit zu widmen. Auf jeden Fall sollte der Chief Scrum Master genug Scrum Sachkenntnis haben, um in der Lage zu sein, Zusammenarbeit zu fördern und andere zu unterstützen und anzuleiten und damit für eine reibungslose Lieferung der Produkte des Projekts zu sorgen.

Neben der Beseitigung von Hindernissen und der Schaffung einer förderlichen Projektumgebung für die Scrum Teams arbeitet der Chief Scrum Master mit dem Chief Product Owner, anderen Scrum Mastern und Product Ownern bei Aktivitäten wie der Erarbeitung der Liste von Komponenten und Ressourcen, die in gemeinsam für alle Teams im Projekt benötigt werden. Er/sie organisiert alles, was über den Bereich eines einzelnen Scrum Teams hinausgeht.

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

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

3.5.2 Program Scrum Master

Der Programm Scrum Master ist ein Vermittler, der sicherstellt, dass alle Projektteams des Programms in einem Umfeld arbeiten, das dem erfolgreichen Abschluss des Projekts förderlich ist. Der Programm Scrum Master leitet an und lehrt jeden, der am Programm beteiligt ist, Scrum Praktiken; gibt Ratschläge an die Scrum Master der individuellen Projekte, kümmert sich um Hindernisse, arbeitet mit dem Scrum Guidance Body bezüglich Zielsetzungen hinsichtlich Qualität, staatlicher Richtlinien, Sicherheit und anderer organisatorischer Schlüsselparameter zusammen darüber hinaus sichert er ab, dass die Scrum-Prozesse während des gesamten Programms effektiv verfolgt werden.

Der Programm Scrum Master spricht sich mit dem Portfolio Scrum Master ab, um sicherzustellen, dass das Programms mit den Zielen des Portfolios kompatibel ist. Sie/er ist darüber hinaus an der Auswahl des jeweiligen Scrum Masters für individuelle Projekte beteiligt und stellt sicher, dass die Vision, die Ziele, die Ergebnisse und die Releases eines individuellen Projektes innerhalb des Programms mit denen des Programms kompatibel sind.

Diese Rolle ist der des Scrum Masters ähnlich, mit dem Unterschied, dass sie den Bedarf des Programms- oder des Geschäftsbereichs statt eines einzelnen Scrum Teams abdeckt.

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.

3.5.3 Portfolio Scrum Master

Diese Rolle ist der des Scrum Masters ähnlich, mit dem Unterschied, dass sie den Bedarf des Portfolios oder des Geschäftsbereichs statt eines einzelnen Scrum Teams abdeckt.

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.

3.6 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 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 zusammen.

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
9.6t Sprint Backlog erstellen	<ul style="list-style-type: none"> • Erstellen des Sprint Backlogs und des Sprint Burndown Charts
10.1 Zu liefernde Ergebnisse erzeugen	<ul style="list-style-type: none"> • Erzeugung der zu liefernden Ergebnisse • Risikoidentifizierung und Einbauen von Maßnahmen zur Risikominderung, falls nötig • Hindernis Log und Abhängigkeiten aktualisieren
10.2 Tägliches Standup durchführen	<ul style="list-style-type: none"> • Burndown Chart, Scrumboard und Impediment Log aktualisieren • Probleme einzelner Mitglieder besprechen und Lösungen zur Motivation des Team suchen • Risiken identifizieren, falls nötig • Change Requests einreichen, falls nötig
10.3 Priorisiertes Produkt Backlog pflegen	<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"> • Vorführung der fertiggestellten zu liefernden Ergebnisse zur Genehmigung durch den Product Owner
11.2 Sprint Rückblick halten	<ul style="list-style-type: none"> • Gelegenheiten zur Verbesserung aus dem laufenden Sprint identifizieren, falls vorhanden und Einigung auf umsetzbare Verbesserungen für den nächsten Sprint
12.2 Projekt Rückblick halten	<ul style="list-style-type: none"> • Teilnahme am Projekt 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

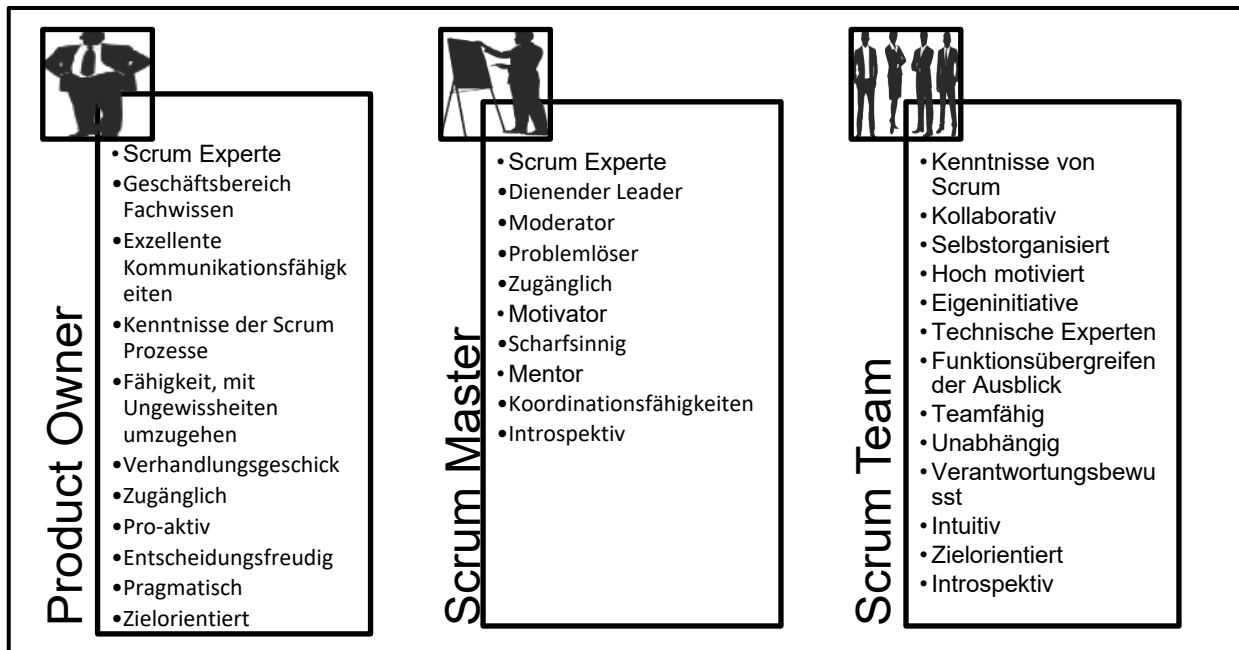


Abbildung 3-3: Wünschenswerte Eigenschaften für die Scrum Kernrollen

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. Ein Hauptvorteil eines Teams von sechs bis zehn Mitgliedern ist, dass Kommunikation und Management normalerweise einfach sind und nur einen minimalen Aufwand darstellen. Allerdings gibt es auch Nachteile. 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 Projekten, Programmen und Portfolios

3.7.1 Definition von Projekt, Programm und Portfolio

- **Projekt**—Unter einem Projekt versteht man ein gemeinschaftliches Unternehmen zur Erstellung von neuen Produkten oder Services oder Ergebnissen, wie sie in der Projekt Visionserklärung definiert sind. Einschränkungen, denen Projekte in der Regel unterliegen, sind Zeit, Kosten, Lieferumfang, Qualität, Personen und die Fähigkeiten der Organisation(en). Das Ziel des Projektteams ist es, die zu liefernden Ergebnisse zu erstellen, wie sie im Priorisierten Produkt Backlog definiert sind.
- **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.
- **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.

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.

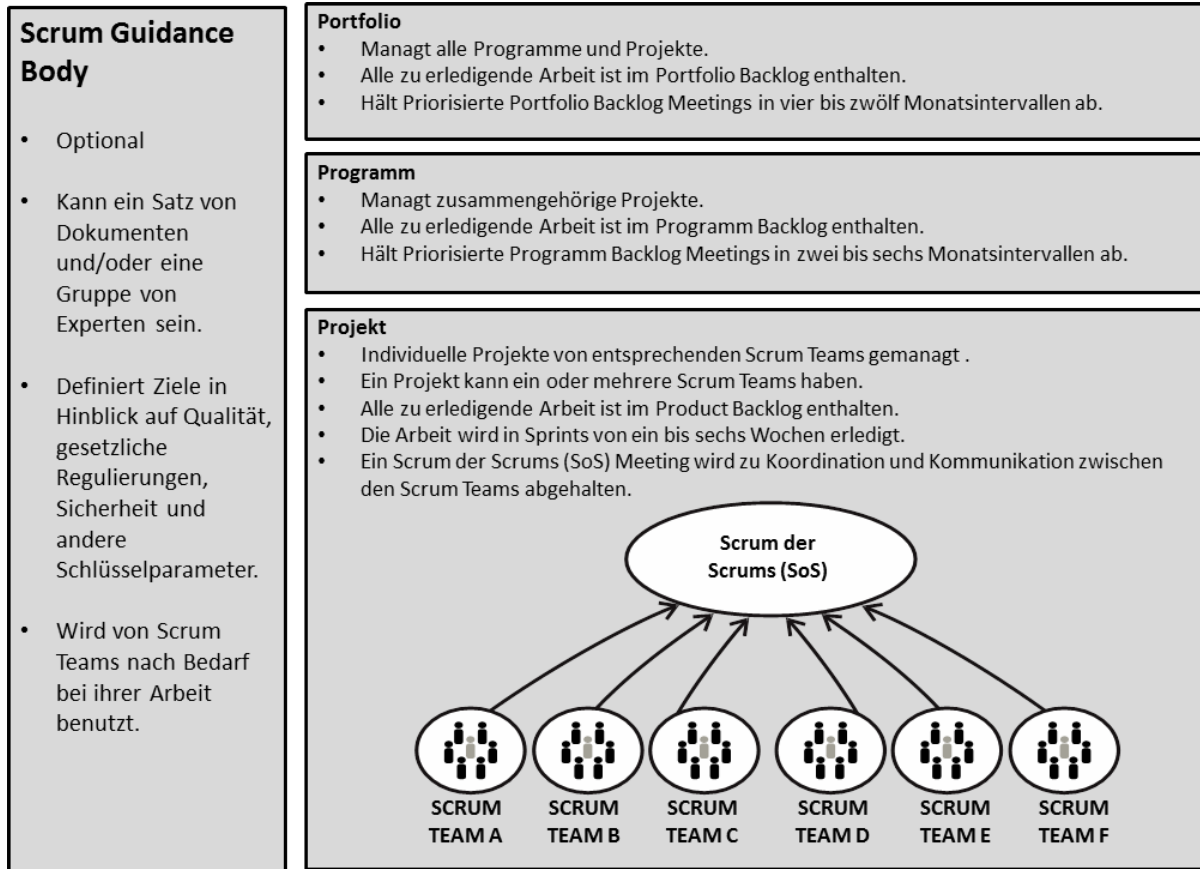


Abbildung 3-4: Scrum quer durch die Organisation für Projekte, Programme und Portfolios

3.7.1.1 Mit Programm und Portfolio Teams zusammen arbeiten

Wenn Scrum zur Verwaltung von Projekten im Kontext eines Programms oder Portfolios angewandt wird, wird stark empfohlen, dass sich an die Scrum Grundsätze aus dieser Publikation gehalten wird. Natürlich ist es verständlich, dass, um die gesamten Programm oder Portfolio Tätigkeiten und gegenseitigen Abhängigkeiten unterzubringen, geringe Anpassungen der Hilfsmittel und der organisatorischen Struktur nötig sein mögen.

Portfolios und Programme haben getrennte Teams mit unterschiedlichen Zielsetzungen. Programm Management Teams zielen darauf ab, Fähigkeiten zu liefern und bestimmte Ziele zu verwirklichen, was dazu beiträgt, die spezifizierten Programmziele zu erreichen. Im Gegensatz dazu muss das Portfolio Team die Ziele der verschiedenen Programme balancieren, um die strategischen Ziele der gesamten Organisation zu erreichen.

3.7.1.2 Kommunikation mit Programm und Portfolio Teams managen

Die Probleme und Angelegenheiten, denen man bei der Scrum Benutzung innerhalb eines Programms oder Portfolios begegnet, sind hauptsächlich mit der Koordination zahlreicher Teams verbunden. Dies kann zum Scheitern führen, wenn man sie nicht sorgfältig managt. Hilfsmittel zur Erleichterung der Kommunikation müssen den Anforderungen der vielen beteiligten Teams eines Programms oder Portfolios Tools angepasst werden. Jedes Scrum Team muss sich nicht nur mit der internen Kommunikation, sondern auch mit der externen Kommunikation mit den anderen Teams und relevanten Stakeholdern des Programms oder Portfolios befassen.

3.7.2 Beteiligung der Stakeholder erhalten

Scrum benötigt die vollkommene Unterstützung der Projekt Stakeholder. Die Verantwortung, die Stakeholder beteiligt zu halten, liegt beim Product Owner. Die folgenden Maßnahmen werden empfohlen, um die Unterstützung und das Engagement der Stakeholder zu behalten:

- Eine effektive Zusammenarbeit und Einbeziehung der Stakeholder am Projekt
- Kontinuierliche Bewertung der Geschäftsauswirkung
- Eine regelmäßige Kommunikation mit den Stakeholdern
- Auf die Erwartungen der Stakeholder eingehen

Der Sponsor ist ein entscheidender Stakeholder—er liefert die Gelder und andere Ressourcen für das Projekt. Sponsoren möchten die finanziellen Endresultate eines Produkts oder einer Dienstleistung verstehen und sind normalerweise mehr am Endergebnis als den individuellen Aufgaben interessiert.

Es ist wichtig, dass die Sponsoren, die das Projekt finanzieren, eine klare Übersicht über folgende Angelegenheiten haben:

- Vorteile der Implementierung von Scrum
- Geplante Fristen und veranschlagte Kosten des Scrum Projekts
- Mit den Scrum Projekten verbundene Gesamtrisiken und die Maßnahme, diese zu mildern
- Erwartete Release Termine und letztendlich zu liefernde Ergebnisse

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
Stakeholder	<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 vermarktaren 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 zugeweilte 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 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.

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.

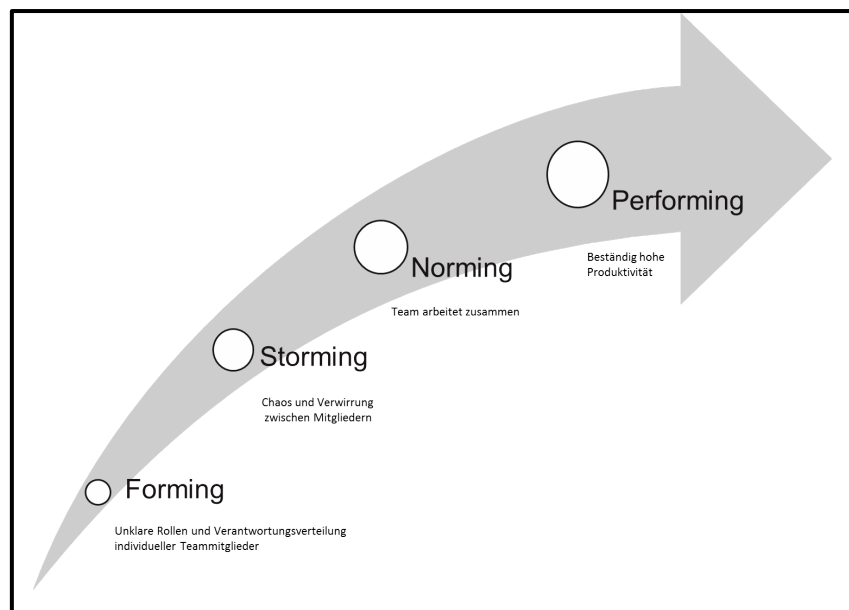


Abbildung 3-4: Phasen der Gruppenentwicklung nach Tuckman

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:

- **Dienender Leader Stil**—Dienende 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 Dienender Leader Stil

Der bevorzugte Führungsstil für ein Scrum Projekt ist der Dienender Leader Stil. Dieser Begriff wurde erstmals von Robert K. Greenleaf im Essay mit dem Titel *The Servant as Leader* (Der Diener als Führer) beschrieben. Nachstehend ein Auszug, in dem er das Konzept erklärt:

Der Dienende Leader *ist* zuallererst Diener. Es beginnt mit dem natürlichen Gefühl, dass man dienen möchte, *zuallererst* dienen. Danach bringt eine bewusste Entscheidung das Erstreben zu führen. So eine Person ist sehr anders als jemand, der zuerst ein *Leader* ist, vielleicht aus dem Bedürfnis heraus, einen ungewöhnlichen Machtdrang zu stillen oder materielle Besitztümer anzuhäufen. Der zuerst-Leader und der zuerst-Diener sind zwei extreme Typen. Zwischen diesen beiden gibt es Schattierungen und Mischungen die Teil der unbegrenzten Vielfalt der menschlichen Natur sind....

Der Unterschied stellt sich in der Sorgfalt des zuerst-Diener heraus, sicherzustellen, dass die höchst-priorären Bedürfnisse anderer Leute erfüllt werden. Der beste Test, schwer durchzuführen, ist: Wachsen die Bedienten als Persönlichkeiten? Werden sie, *während ihnen gedient wird*, gesünder, klüger, freier, selbstständiger, eher dazu bereit selbst Diener zu werden? *Und* was ist der Effekt auf die am wenigsten privilegierten in der Gesellschaft? Werden sie davon profitieren oder wenigstens nicht weiter benachteiligt? (Greenleaf 1970, 6)

Die Schriften von Greenleaf weiter ausführend, identifiziert Larry Spears zehn Merkmale, die jeder effektive Dienende Leader haben sollte:

1. **Zuhören**—Von Dienenden Leaders 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 Dienende 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 des Dienenden Leaders. Dies ermöglicht ihm, Angelegenheiten in Bezug auf Ethik, Macht und Werte besser zu verstehen und zu integrieren.
5. **Überzeugung**—Dienende 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 Dienenden Leaders.
7. **Voraussicht**—Sein intuitiver Verstand ermöglicht dem Dienenden Leader, 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. Dienende Leader ziehen Überzeugung der Kontrolle vor um sicherzustellen, dass sie das Vertrauen anderer in der Organisation gewinnen.
9. **Verpflichtung zum Wachstum anderer**—Dienende 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**—Dienende 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) Dienende 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.

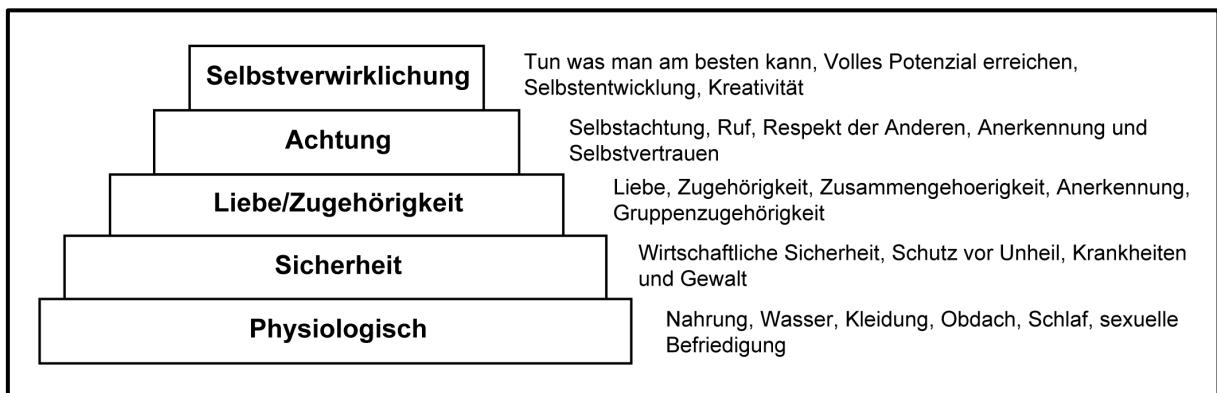


Abbildung 3-5: Hierarchie der Bedürfnisse Theorie nach Maslow

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.

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 Stakeholder 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 Stakeholder 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 Stakeholder.

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 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 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 Stakeholder 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.

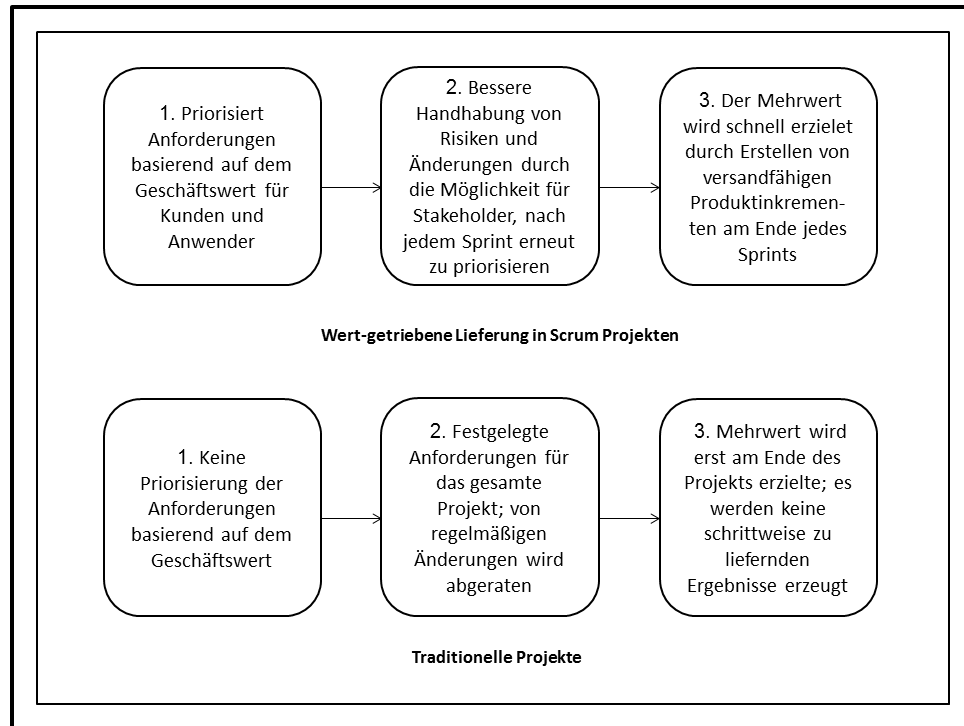


Abbildung 4-1: Lieferung von Mehrwert in Scrum versus traditionellen Projekten

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 Stakeholder

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 Stakeholder 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 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 Stakeholder 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

4

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 Stakeholder 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 pflegen*. 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 *Projekt Rückblick halten* Prozessen realisiert.

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

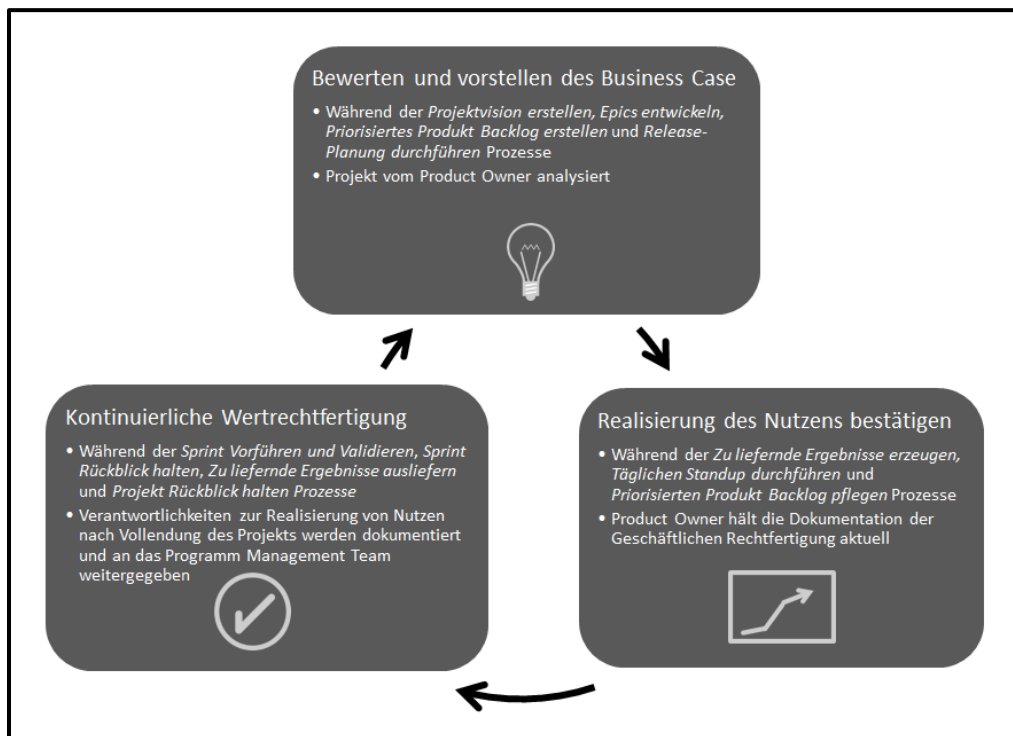


Abbildung 4-3: Geschäftliche Rechtfertigung und der Projekt Lebenszyklus

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 miteinander berechnet.

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?

3. Projekt A hat einen IRR von 15% und wird in 5 Jahren abgeschlossen sein.
4. 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 Stakeholder (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. Wertstromanalyse

Wertstromanalyse nutzt Flussdiagramme, um den zur Vervollständigung eines Prozesses erforderlichen Informationsfluss zu illustrieren. Sie kann dazu verwendet werden, einen Prozess durch Identifizierung der nicht-wertschöpfenden Elemente zu straffen.

2. 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.

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

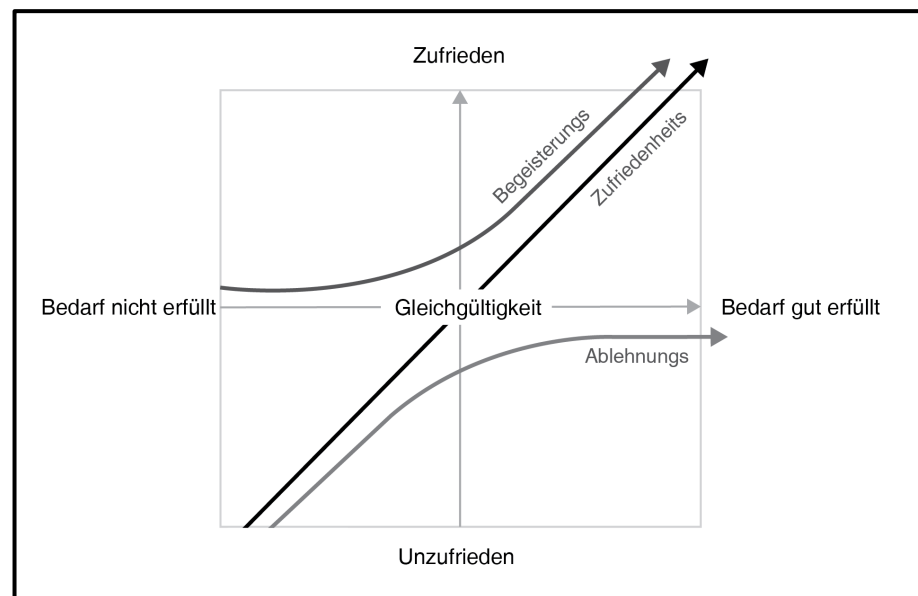


Abbildung 4-4: Kano-Analyse

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 höchstpriorien in der Priorisierungsrangliste sein.

4.5.4 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.

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 Stakeholder.

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
Kostenentwicklungsindex	CPI	EV / AC
Prozent Abgeschlossen	% Complete	(EV / BAC) x 100
Erwartete Kosten zur Fertigstellung <ul style="list-style-type: none"> • Kalkulierte Annahmen nicht gültig • Derzeitige Abweichungen sind atypisch • Derzeitige Abweichungen sind typisch 	EAC	4. AC + ETC 5. AC + BAC - EV 6. BAC / CPI
Erwartete Restkosten zur Fertigstellung	ETC	EAC - AC
Gesamtkostenabweichung bei Abschluss	VAC	BAC - EAC

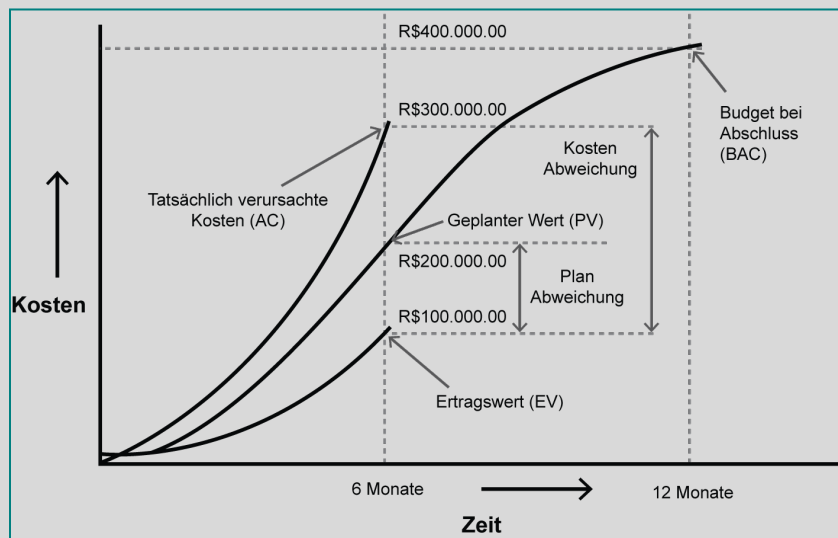
Tabelle 4-1: Ertragswert Formeln

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 4-5 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.

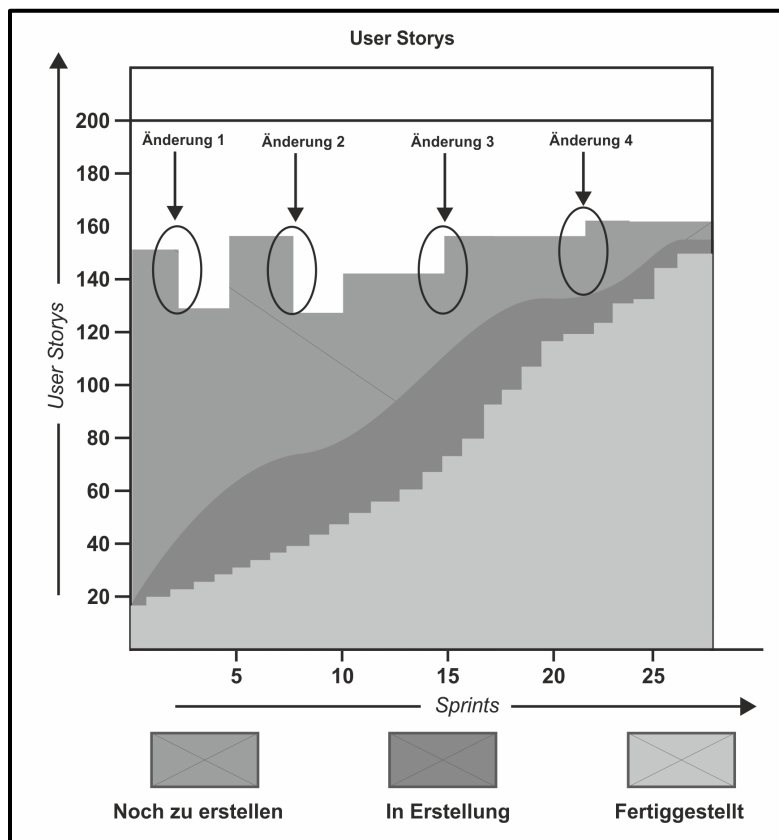


Abbildung 4-5: Beispiel Kumulatives Flussdiagramm (CFD)

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
Stakeholder	<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 Stakeholder
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 Stakeholder 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 Stakeholder 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 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 Stories, 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 Stories 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.

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 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 des *Priorisiertes Produkt Backlog pflegen* Prozesses in zukünftigen Sprints vervollständig zu werden. Die Ablehnung einiger individuell zu liefernden Ergebnisse und deren zugehöriger User Storys ist keine Ablehnung des Endprodukts oder Produkt-Inkrement. 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.

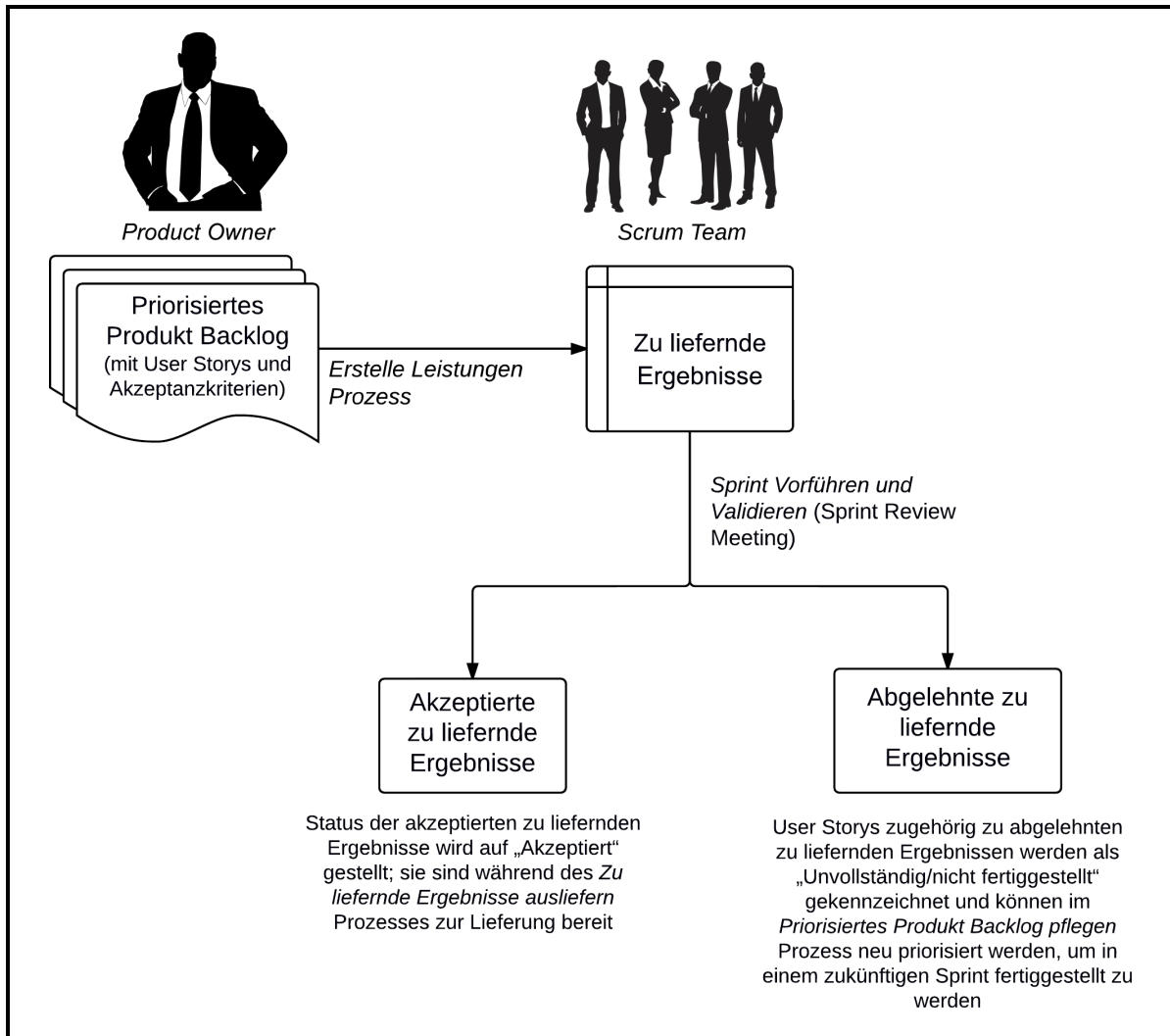


Abbildung 5-1: Projekt-Inkrement Flussdiagramm

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. Über verschiedene 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.“

Akzeptanzkriterium:

- 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 Done

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 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 wird normalerweise im Scrum Guidance Body festgesetzt und dokumentiert.

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.

5.4.3 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

	<ul style="list-style-type: none"> • 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.

5.4.4 Akzeptieren oder Ablehnen eines Priorisierten Produkt Backlog Artikels

Zum Ende jeder Iteration beteiligen sich die betreffende Geschäftseinheit und die Stakeholder an einem Sprint Review Meeting, in dem das Produkt-Inkrement dem Product Owner, Sponsor, Kunden und Anwendern vorgeführt wird. Von allen 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 Stakeholder 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 Stakeholder 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 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 eine 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.

5.5.2 Qualitätskontrolle und Qualitätssicherung

Die Qualitätskontrolle bezieht sich auf die Durchführung der geplanten Qualitätsmaßnahmen, die vom Scrum Team bei der Erstellung von versandfähigen abzuliefernden 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 Rückblick-

Meetings besprechen die Team-Mitglieder ihre Erfahrungen. Diese Lehren dienen als Input zur kontinuierlichen Verbesserung und tragen zur Optimierung der anhaltenden Qualitätskontrolle bei.

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 Produkts, 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.

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.

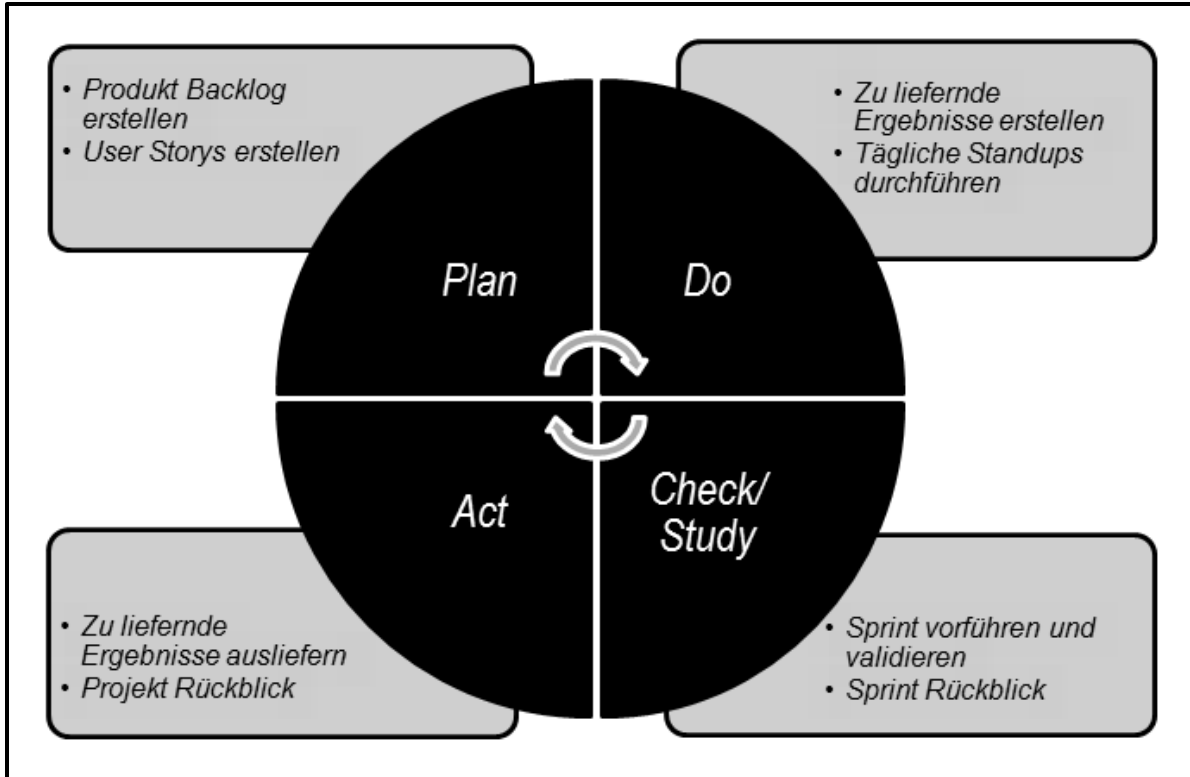


Abbildung 5-2: PDCA Zyklus in Scrum

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
Stakeholder	<ul style="list-style-type: none"> • Überprüft und akzeptiert die zu liefernden Ergebnisse und das Endprodukt
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

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 Stakeholder 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 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 ist 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 Stakeholder 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) Stakeholder (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 Stakeholder 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 den Toleranzlevel des Product Owner hinausgehen, brauchen eventuell die Genehmigung der betreffenden Stakeholder, 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 pflegen* besprochen und genehmigt. *Genehmigte Change Requests* werden dann zusammen mit den anderen Produkthanforderungen 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.

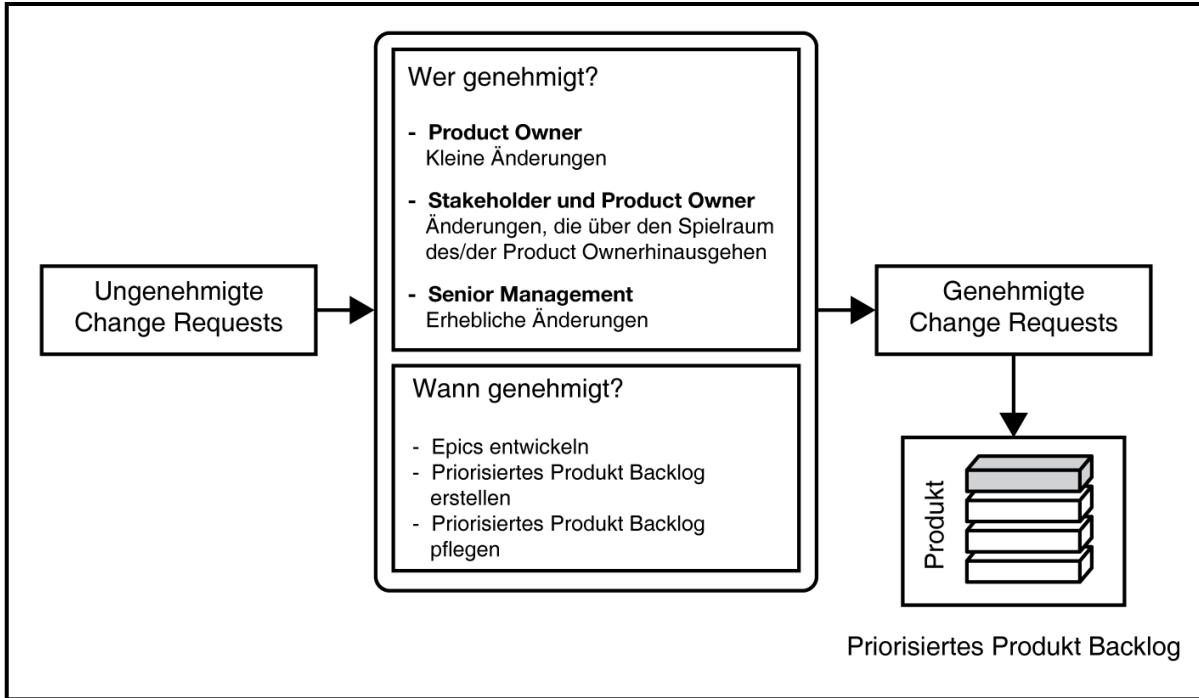


Abbildung 6-1: Beispiel genehmigter Änderungsprozess

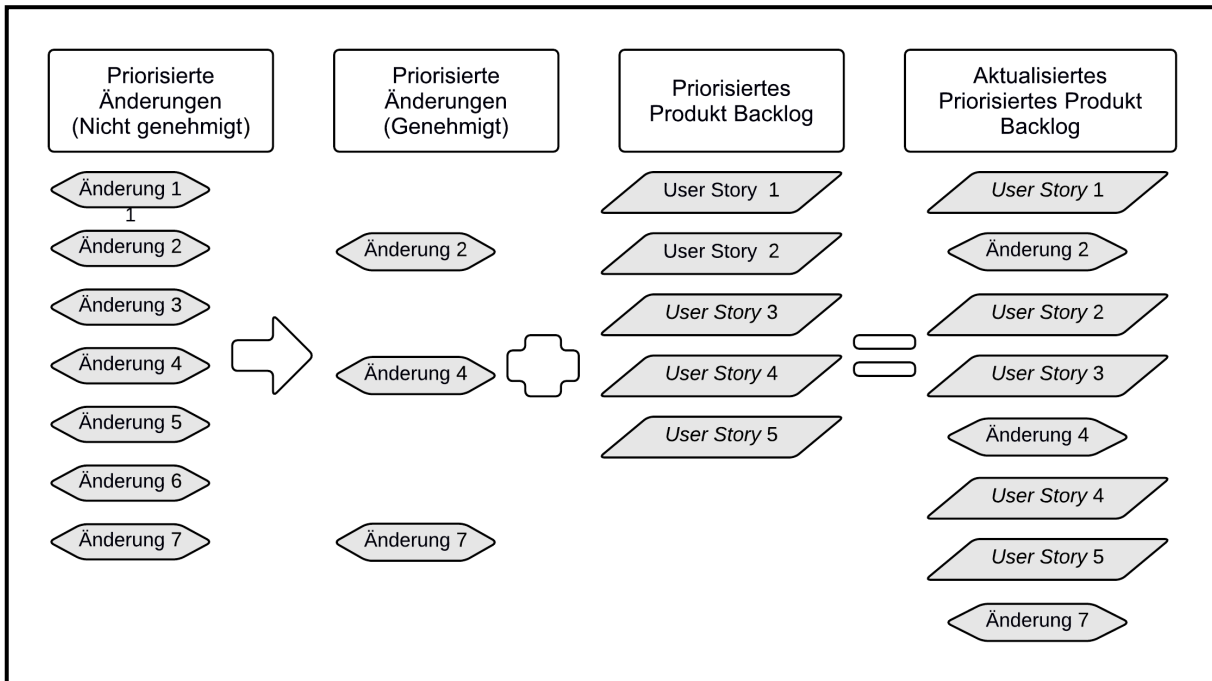


Abbildung 6-2: Aktualisieren des Priorisierten Produkt Backlogs mit genehmigten Änderungen

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

Scrum vereinfacht Flexibilität durch *Transparenz*, *Inspektion* und *Adaptierung*, um das bestmögliche Geschäftsergebnis zu erzielen. 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.

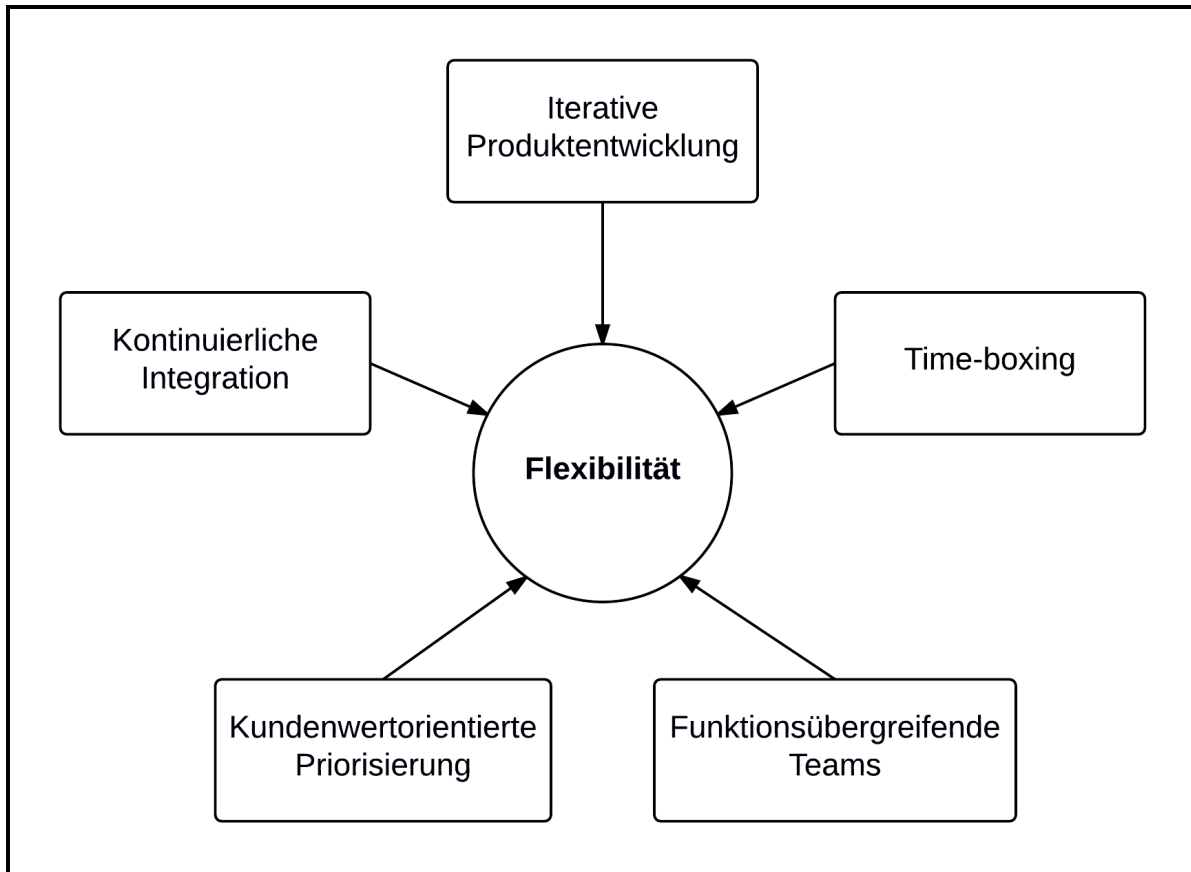


Abbildung 6-3: Scrum Merkmale um Flexibilität zu erreichen

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. Stakeholder

Projekt Stakeholder—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 Stakeholder Change Requests einreichen, während sie die zu liefernden Ergebnisse der Prozesse *Sprint Vorführen und Validieren*, *Sprint Rückblick halten*, oder *Projekt Rückblick halten* überprüfen. Alle Change Requests werden dem Priorisierten Produkt Backlog hinzugefügt, sobald sie genehmigt wurden. Abbildung 6-4 veranschaulicht einige Gründe für Stakeholder, einen Change Request Prozess einzuleiten.

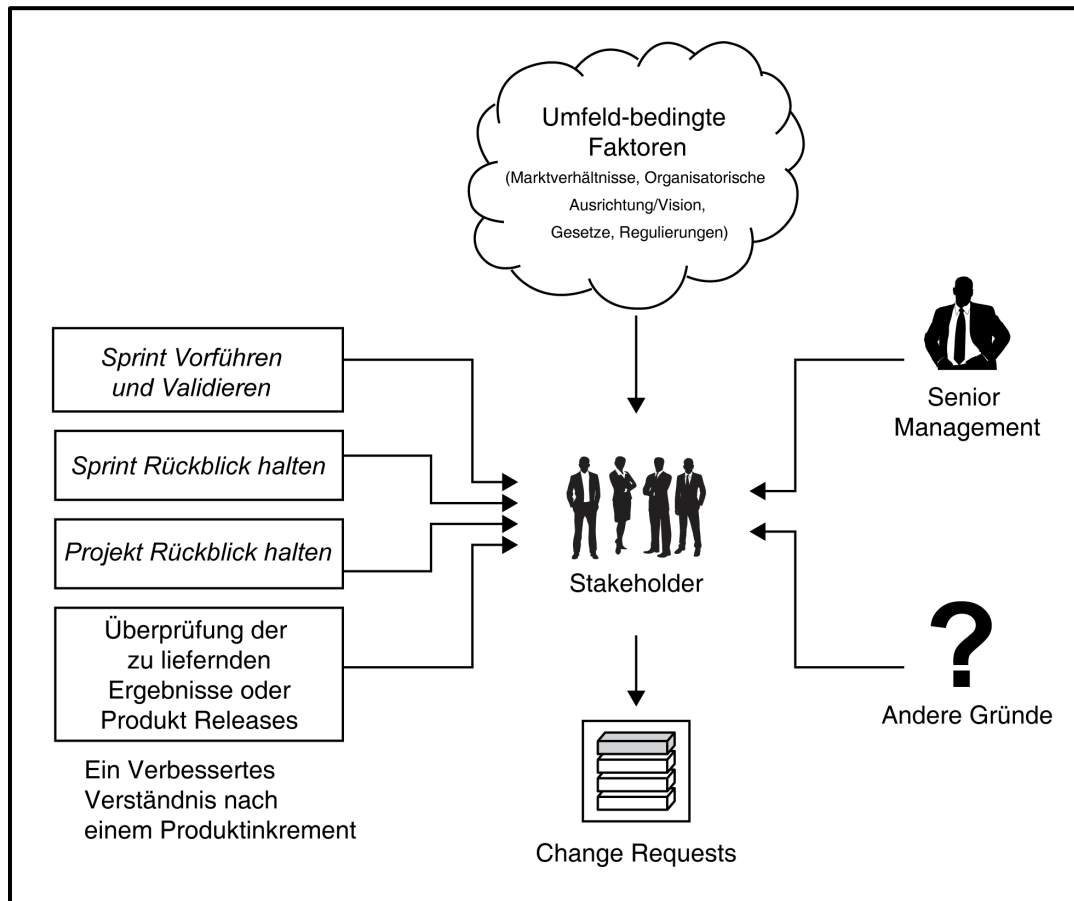


Abbildung 6-4: Beweggründe der Stakeholder, Änderungen anzufragen

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 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 *Tägliches Standup Meeting* oder *Sprint Retrospektions-Meetings* erzeugt. Abbildung 6-5 veranschaulicht einige Gründe, warum das Scrum Kernteam Change Requests einreichen mag.

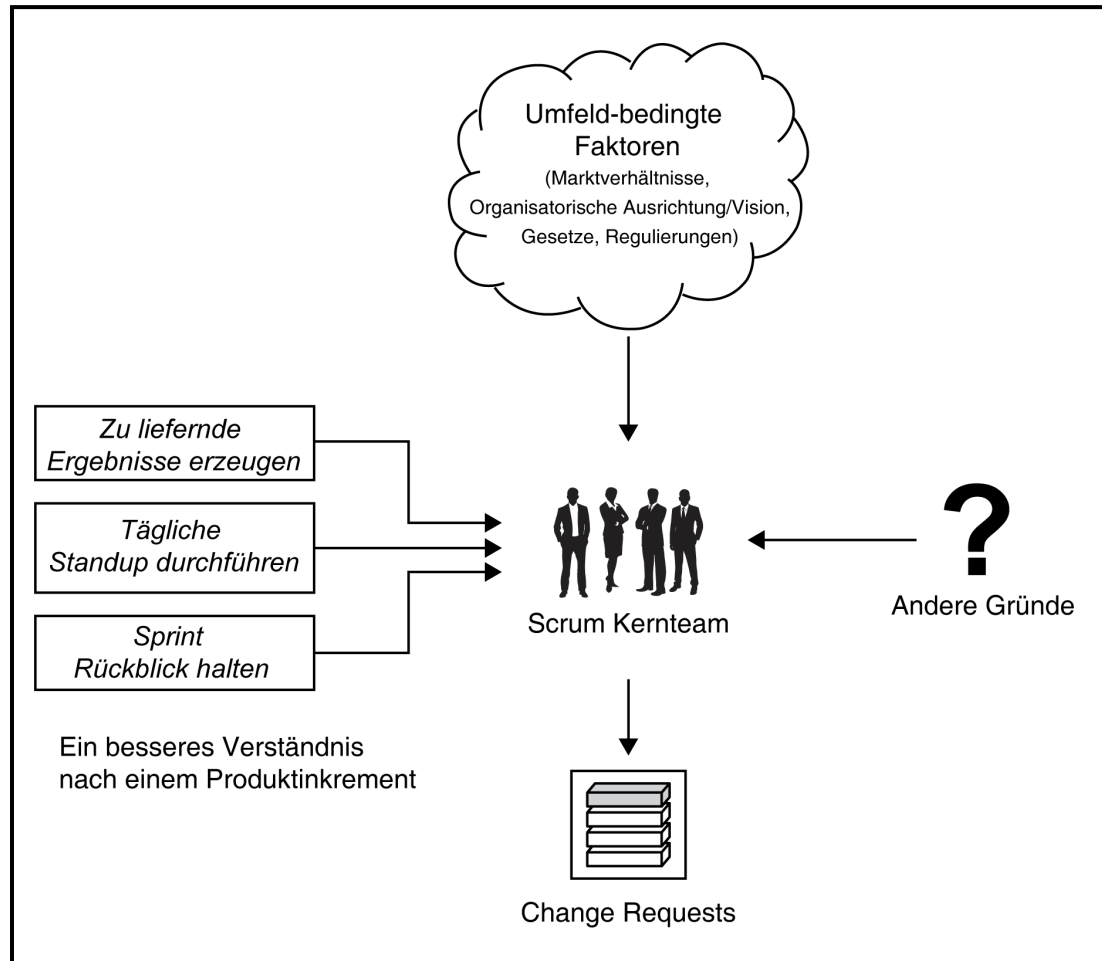


Abbildung 6-5: Beweggründe des Scrum Kernteams, Änderungen anzufordern

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.

6.4.2.2 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 sechs Wochen festzusetzen. 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. Time-boxing hilft, den Fortschritt eines Projekts abzumessen und erleichtert dem Team, Prozesse oder Ansätze zu erkennen, die modifiziert werden sollten.

Dass Sprints time-boxed sind trägt einen großen Teil dazu bei, Abgabetermine einzuhalten und eine höhere Produktivität zu erreichen. 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. Fehler oder Probleme können frühzeitig identifiziert und schnell behoben werden.

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 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 versandfähig und bereit zur Vorführung vor Product Ownern und/oder anderen 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 pflegen* 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 pflegen*—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 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).

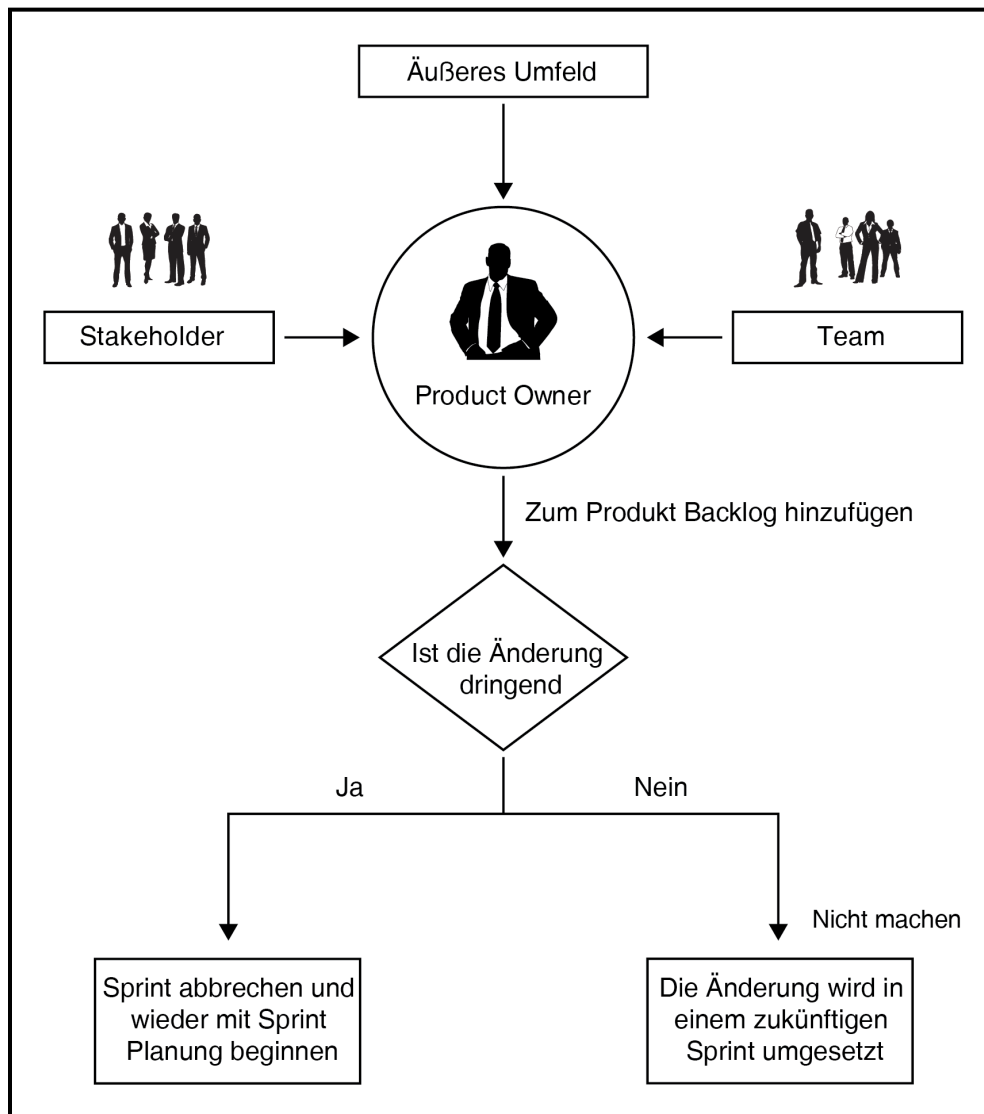


Abbildung 6-6: Integration von Änderung in Scrum

Es gibt nur eine Ausnahme dieser Regel, den Umfang eines Sprints 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 Stakeholder Änderungen und Anforderungen während der dafür vorgesehenen Prozesse *Epic(s) entwickeln*, *Priorisiertes Produkt Backlog erstellen* und *Priorisiertes Produkt Backlog pflegen*, da sie sich bewusst sind, dass Änderungen während eines Sprints nicht erlaubt sind und ein Sprint nicht mehr als 1 und 6 Wochen dauert.

6.5.1.1 Der erwartete Einfluss einer Änderung 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.

Wenn Projekt Anforderungen generell stabil sind und größere Änderungen in der näheren Zukunft nicht erwartet werden, kann die Länge eines Sprints auf 4 bis 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, dies mittels der Prozesse *User Storys Erstellen*; *User Storys abschätzen* und *User Storys zusagen*, *Aufgabenidentifizieren*, *Aufgaben abschätzen* und andere verwandte Prozesse, die in jedem Sprint durchgeführt werden, zu tun.

Wenn allerdings Produktanforderungen nicht deutlich definiert wurden oder wenn wichtige Änderungen in der nahen Zukunft erwartet werden, kann die Sprint-Länge erheblich kürzer sein, 4 Wochen oder weniger. Dies bietet dem Scrum Team die nötige Stabilität an kurzen Sprints zu arbeiten und Ergebnisse zu liefern, welche vom Product Owner und den 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.

Um aus einem Scrum Projekt die maximalen Vorteile zu ziehen, wird empfohlen, die Sprints zeitlich auf nicht mehr als 4 Wochen zu veranschlagen, es sei denn, es sind Projekte mit sehr stabilen Anforderungen, in denen Sprints sich bis zu 6 Wochen strecken können.

Abbildung 6-7 stellt die Auswirkung erwarteter Änderungen auf die Sprint-Länge dar.

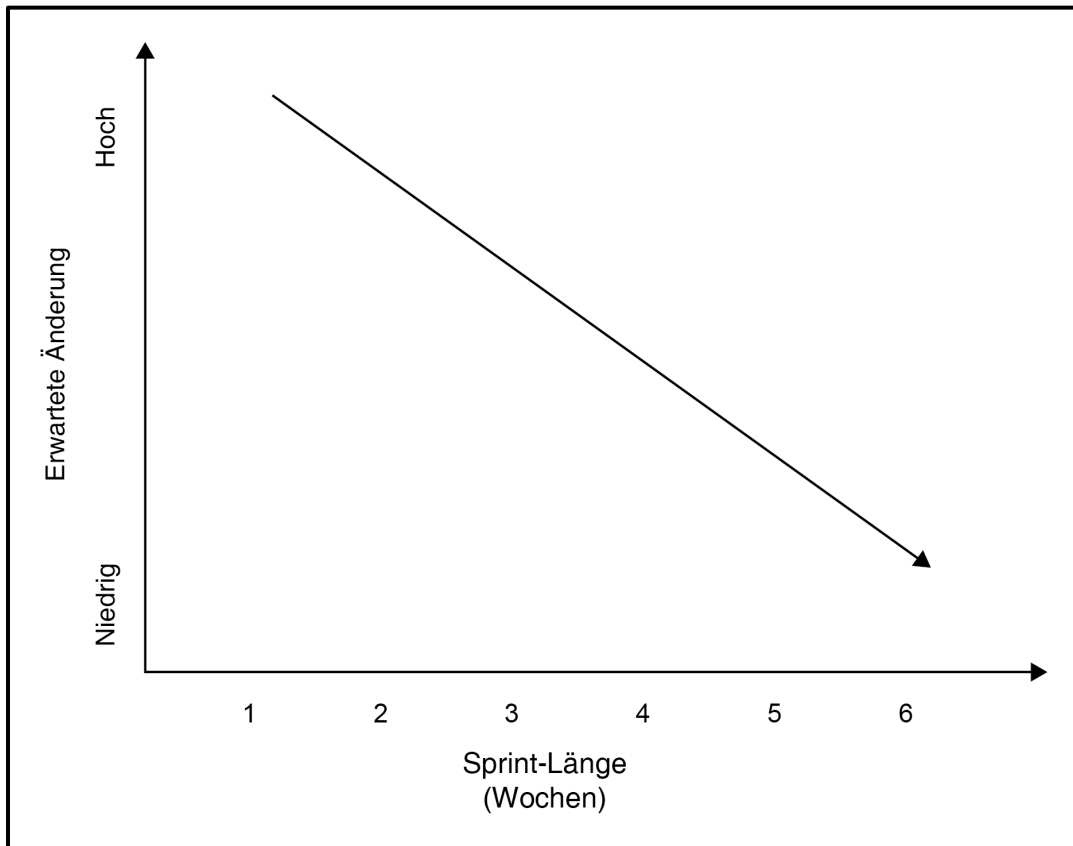


Abbildung 6-7: Auswirkung der erwarteten Änderungen auf die Sprint-Länge

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 eine 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 zuzusagen. 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 pflegen“ 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 Stakeholder 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 erstellen* 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 pflegen* 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 Stakeholder zu begrenzen. Zu viele Teilnehmer zu haben birgt die Gefahr, die Effizienz des gesamten Meetings zu mindern. Der Product Owner sollte nur die Stakeholder 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.

Pflegeaktivitäten 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 Stakeholder 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 Stakeholder 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 Stakeholder (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 Stakeholder (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 Programme gehandhabt werden können.

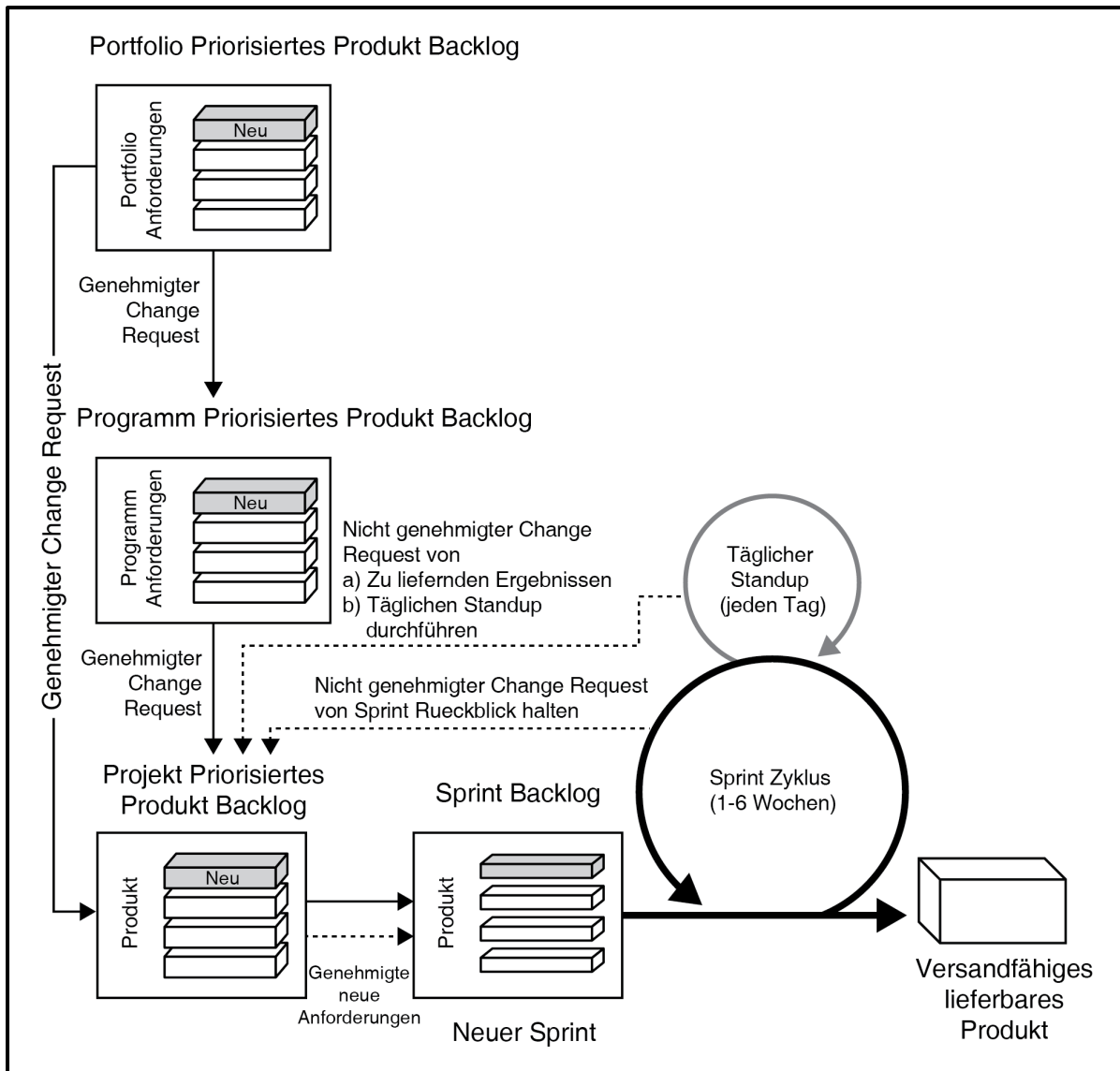


Abbildung 6-8: Änderungen im Portfolio und Programm einbinden

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
Stakeholder	<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 Stakeholder 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</i> und <i>Täglichen Standup durchführen</i> vor

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 Stakeholder 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 Stakeholder Change Requests zu dem Priorisierten Produkt Backlog hinzufügen. Product Owner und Stakeholder 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 Stakeholder 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 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 versandfähigen zu liefernden Ergebnisse können vom Product Owner und anderen 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 Stakeholder 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

Stakeholder 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 Stakeholdern zu verstehen. Risikoverhalten wird von diesen drei Faktoren beeinflusst:

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

Grundsätzlich bestimmt das Risikoverhalten von 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 Stakeholder 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 Stakeholdern oder ihre Haltung gegenüber von Risiken zu messen. Es definiert die Grenzen und Bereitschaft von Stakeholdern, Risiken zu akzeptieren. Die Nutzenfunktion hat drei Kategorien:

1. Risikoscheu: Der Stakeholder ist nicht bereit, ein Risiko zu akzeptieren, egal welcher Nutzen oder welche Gelegenheit dabei erwartet wird.
2. Risikoneutral: der Stakeholder 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 Stakeholder nicht besorgt sein, wenn eines der Szenarien riskanter als das andere ist.
3. Risikosuchend: Der Stakeholder 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 Stakeholder 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 pflegen* und *Sprint vorführen 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 Stakeholder 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.

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 Stakeholder, 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).

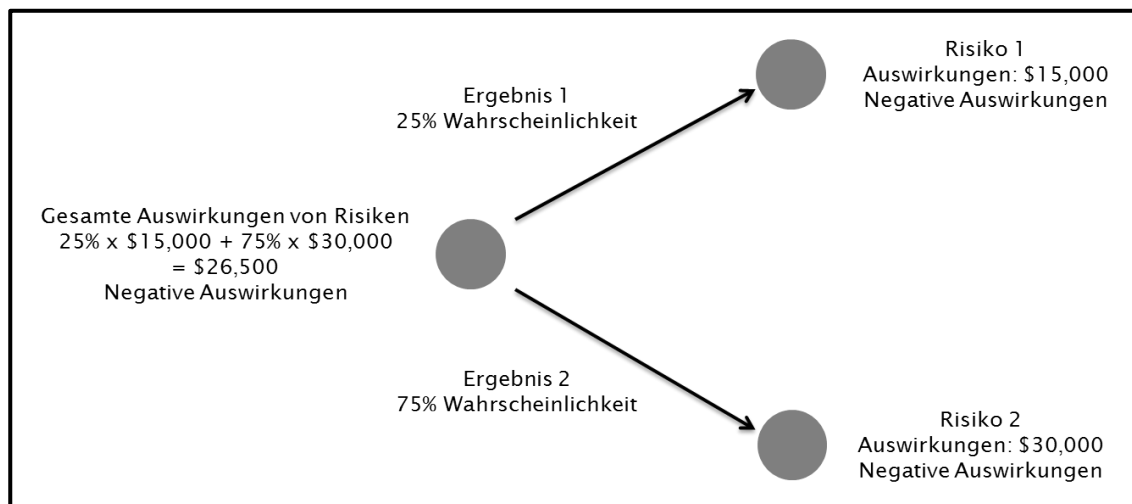


Abbildung 7-1: Beispiel für einen Wahrscheinlichkeitsbaum

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.

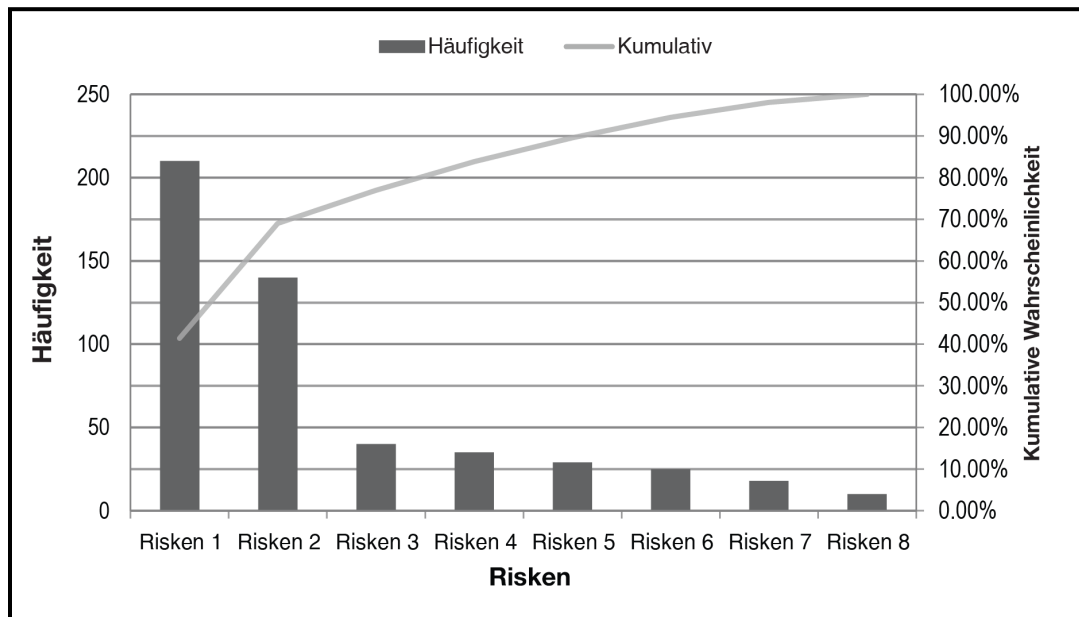


Abbildung 7-2: Beispiel eines Pareto-Diagramms

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.

Die Kennzahl für die Schwere des Risikos bei einer Risikowahrscheinlichkeit von 50% und einer Auswirkungsbewertung von 0,6 würde beispielsweise 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						

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 Risikenliste. (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 entscheiden kann, sie durch eine gezielte Risikomaßnahme während des Sprints zu mindern, werden ausgewählt.
3. Eine Liste von User Storys wird im Priorisierten Produkt Backlog erstellt; diese sind nach ihrem Wert geordnet (Der Wert jeder einzelnen User Story kann aufgrund der erwarteten Investmentrendite eingeschätzt werden)
4. Die Listen der Schritte 2 und 3 werden kombiniert und nach ihrem Wert priorisiert, um zu einem aktualisierten Priorisierten Produkt Backlog zu gelangen.

Abbildung 7-4 verdeutlicht den Ablauf der Risikopriorisierung.

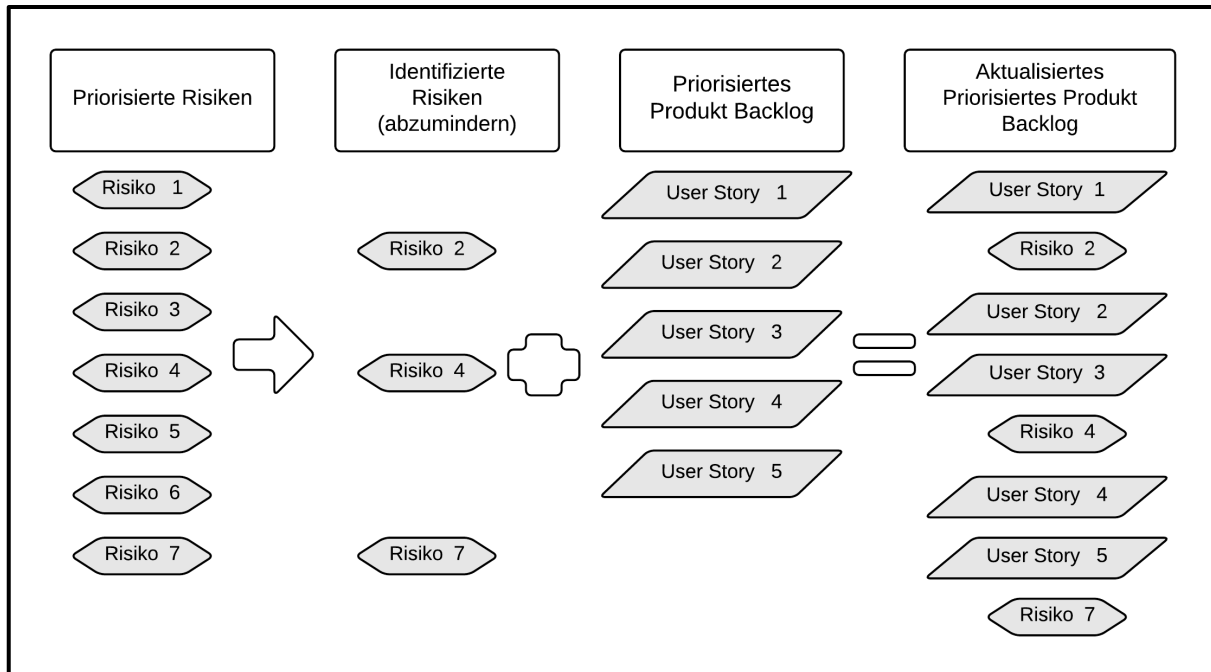


Abbildung 7-4: Ablauf der Risikopriorisierung

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 Risikobasierter Spike

Ein Konzept, welches für das Identifizieren von Risiken nützlich sein kann, ist das des Risikobasierten Spikes. Spikes sind im Grunde ein Experiment, welches Forschung und Prototyping verwendet, um potentielle 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.

7

7.4.5 Risikokommunikation

Da die Stakeholder ein Interesse am Projekt haben, ist es wichtig, mit ihnen im Bezug auf Risiken zu kommunizieren. Die Informationen, welche den 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.

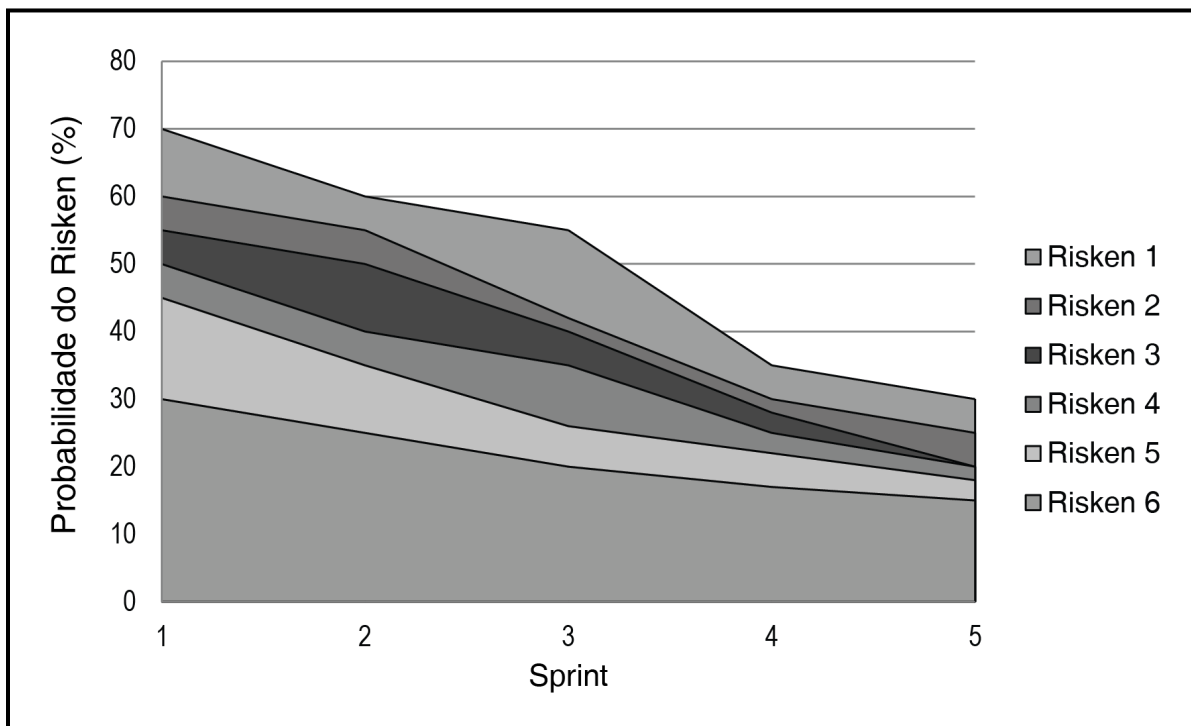


Abbildung 7-5: Beispiel eines Risiko-Burndown-Charts

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 Stakeholder 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 Scrum Meetings 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 potentiell 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 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 Stakeholdern und den Projektteams mitteilen müssen. In manchen Fällen, muss das Programmteam die Verantwortung für eines spezifisches Risiko übernehmen.

Abbildung 7-6 demonstriert, wie Risiken innerhalb eines Scrum Flows sowohl bei Portfolios als auch bei Programmen gehandhabt werden können.

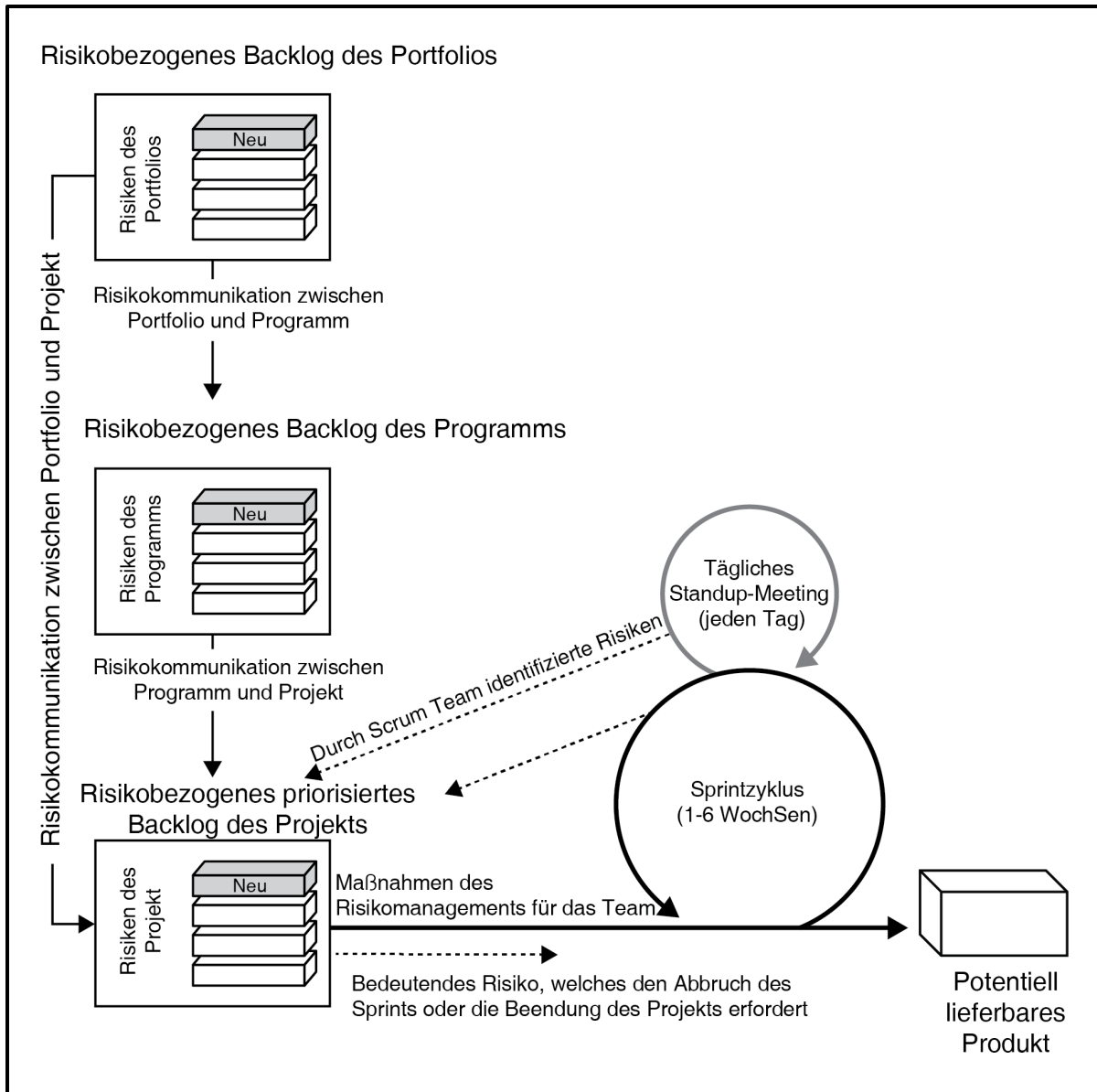


Abbildung 7-6: Handhabung von Risiken in Portfolios und Programmen

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 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 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.
Stakeholder	<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 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 7-7-1: 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 Stakeholder 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 Stakeholder 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. Darüber hinaus ist es ebenfalls von Bedeutung zu verstehen, dass, obwohl alle Vorgänge einzeln im *SBOK™* definiert sind, diese nicht notwendigerweise auch nacheinander oder von einander getrennt ablaufen. Mitunter ist es nützlicher, einige Vorgänge, 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.

Die Abbildung 8-1 bietet einen Überblick über die Vorgänge der Initiierungsphase, welche im Folgenden gelistet sind:

8.1 Projektvision erstellen—In diesem Prozess wird die wirtschaftliche Begründung für das Projekt herangezogen, um eine Projektvision zu erstellen, die als Inspiration und Richtschnur für das gesamte Projekt dient. Dabei wird auch der Product Owner identifiziert.

8.2 Scrum Master und Stakeholder identifizieren—In diesem Prozess wird der Scrum Master anhand von spezifischen Auswahlkriterien bestimmt.

8.3 Scrum Team bilden—In diesem Prozess werden die Mitglieder des Scrum Teams bestimmt. 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 Ausarbeitung von Epic(s). Zur Erstellung von Epic(s) können Treffen zwischen den Anwender-Gruppen abgehalten werden.

8.5 Priorisiertes Produkt Backlog erstellen—In diesem Prozess werden die Epic(s) verfeinert, ausgearbeitet und schließlich priorisiert, um für das Projekt ein priorisiertes Produkt Backlog zu erschaffen. Dabei werden auch die Done-Kriterien festgelegt.

8.6 Release-Planung durchführen—In diesem Prozess überprüft das Scrum Kernteam die User Storys des priorisierten Produkt Backlog um eine Release Planung zu entwickeln, welche im Kern eine stufenweise Lieferplanung darstellt, welche den Projekt Stakeholdern mitgeteilt werden kann. Hierbei wird auch die Länge der Sprints festgelegt.

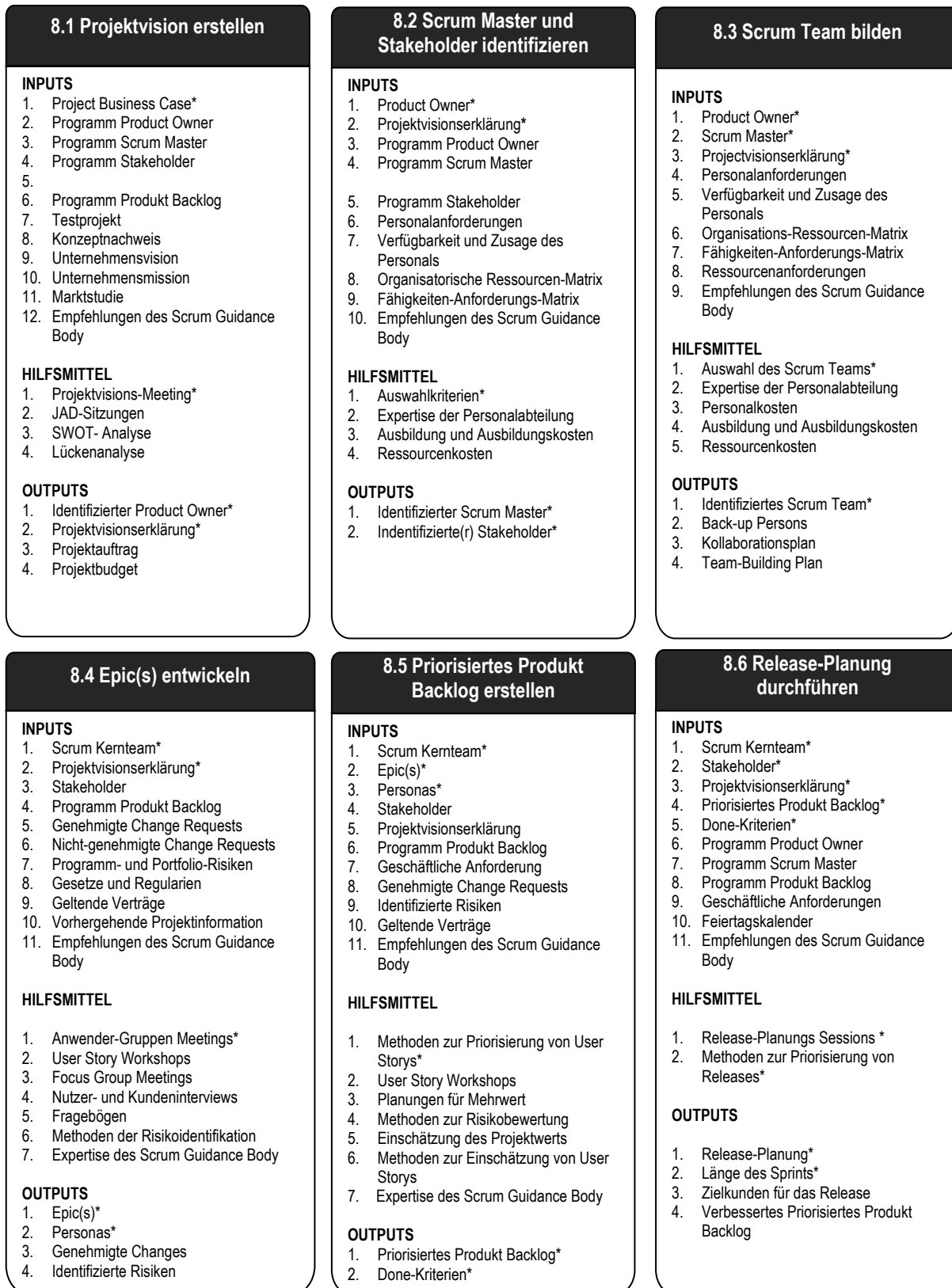


Abbildung 8-1: Überblick über die Initiierungsphase

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

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

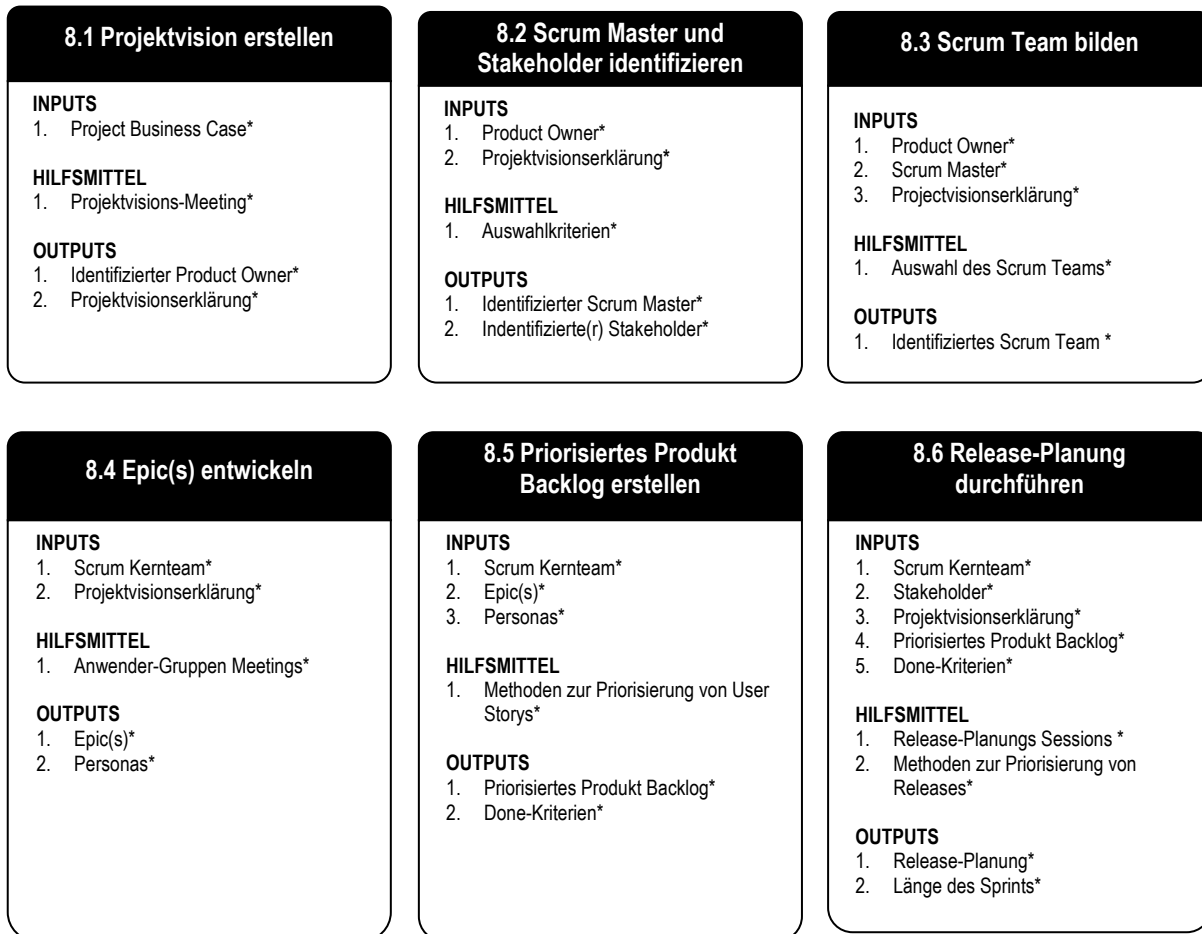


Abbildung 8-2: Überblick über die Initiierungsphase (Kernpunkte)

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

8.1 Projektvision Erstellen

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

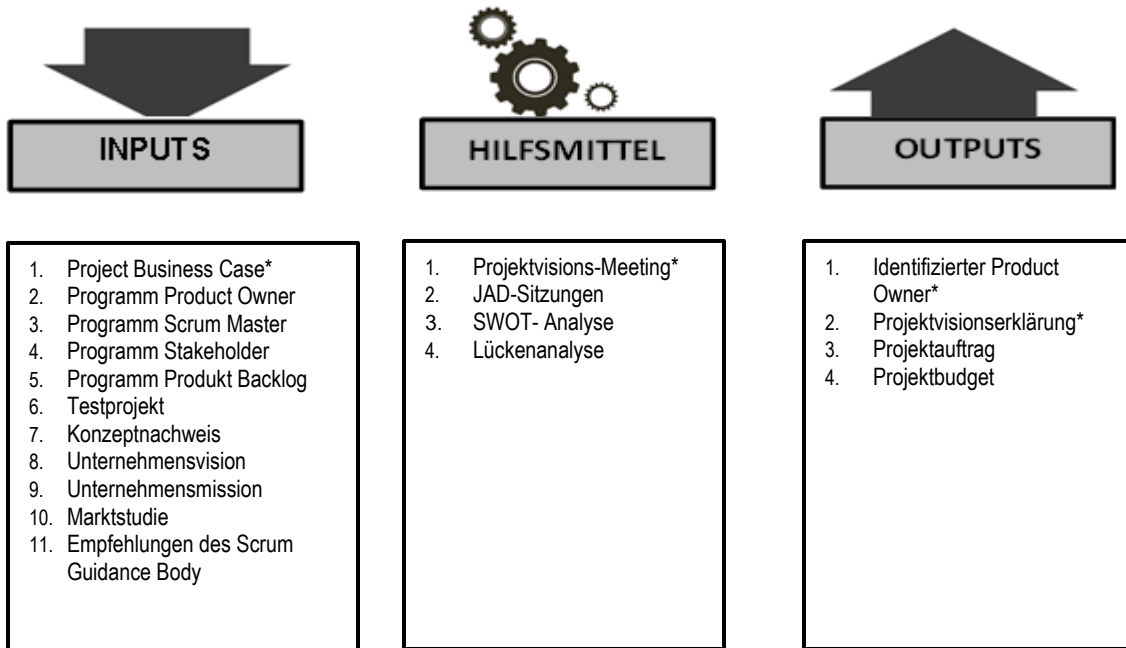


Abbildung 8-3: Projektvision erstellen–Inputs, Hilfsmittel und Outputs

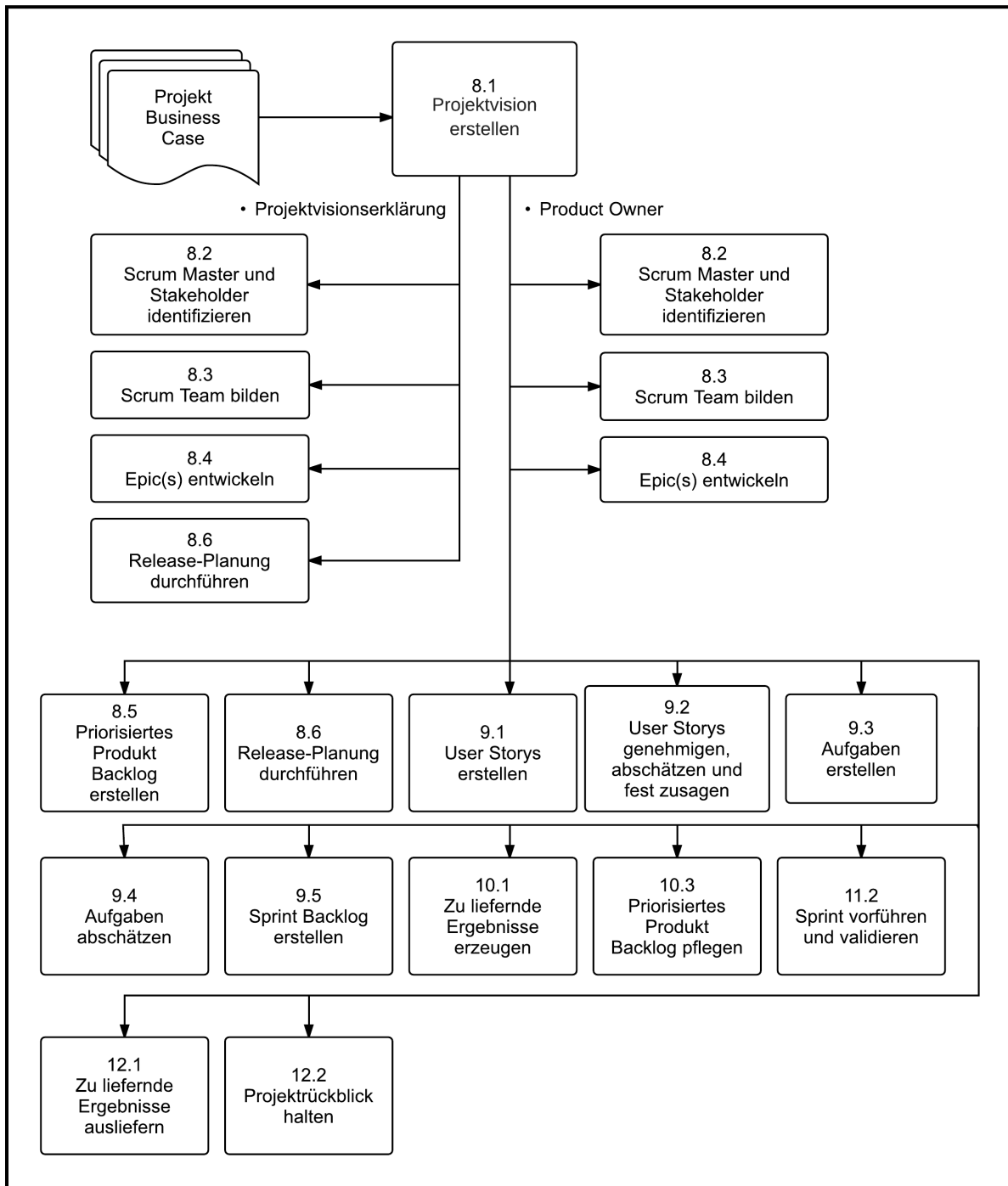


Abbildung 8-4: Projektvision erstellen–Datenflussdiagramm

8.1.1 Inputs

8.1.1.1 Projekt Business Case*

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.

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

8

8.1.1.2 Programm Product Owner

Beschrieben in Abschnitt 3.7.3.1.

8.1.1.3 Programm Scrum Master

Beschrieben in Abschnitt 3.7.3.2.

8.1.1.4 Programm Stakeholder

Programm Stakeholder ist ein Sammelbegriff, der Kunden, Anwender und Sponsoren eines Programms umfasst. Jede dieser Gruppen beeinflusst die Projekte des Programms während der gesamten Projektentwicklung. Die Programm Stakeholder können darüber hinaus dabei helfen, die Projektvision zu definieren und liefern Richtlinien für den Geschäftswert.

Die Programm Stakeholder sprechen sich mit den Portfolio Stakeholdern ab, um die Kompatibilität des Programms mit den Zielen des Portfolios zu sichern. Sie sind darüber hinaus an der Auswahl der jeweiligen Stakeholder für individuelle Projekte beteiligt und sichern, dass die Vision, die Ziele, die Ergebnisse und die

Veröffentlichungen eines individuellen Projektes innerhalb des Programms mit denen des Programms kompatibel sind.

8.1.1.5 Programm Produkt Backlog

Der Programm Product Owner entwickelt das Programm Produkt Backlog, welches eine priorisierte Liste von high-level geschäftlichen und Projekt-Anforderungen enthält, welche bevorzugt in der Form von großen Program Backlog-Einträgen notiert werden. Diese werden später durch den Product Owner des jeweiligen individuellen Projekts weiterentwickelt, wenn diese priorisierte Produkt Backlogs für Ihre Projekte erstellen. Diese priorisierten Produkt Backlogs besitzen deutlich kleinere, aber detaillierte User Storys durch die individuellen Scrum Teams abgeschätzt und zugesagt werden.

Das Programm Produkt Backlog wird beständig durch den Programm Product Owner gepflegt, um abzusichern, dass neue geschäftliche Anforderungen hinzugefügt werden und existierende Anforderungen korrekt dokumentiert und priorisiert sind. Dies garantiert, dass die wichtigsten Anforderungen für die Lösung der Probleme des Programms entsprechend hoch und die restlichen entsprechend niedriger priorisiert sind.

Das Programm Produkt Backlog ermöglicht einen allgemeineren Blick auf alle Projekte, die Teil des Programms sind. Es kann daher eine wesentliche Anleitung im Bezug auf Projektziele, Projektumfang, und erwartete Geschäftsvorteile bereitstellen.

8.1.1.6 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.

8.1.1.7 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.

8.1.1.8 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.

8.1.1.9 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.

8.1.1.10 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.

8.1.1.11 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 Stakeholder, des Programm Product Owners, und des Programm Scrum Masters. Es hilft bei der Identifizierung von Geschäftsumfeld, von geschäftlichen Anforderungen und den Erwartungen der Stakeholder, 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 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 Stakeholder, 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 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.

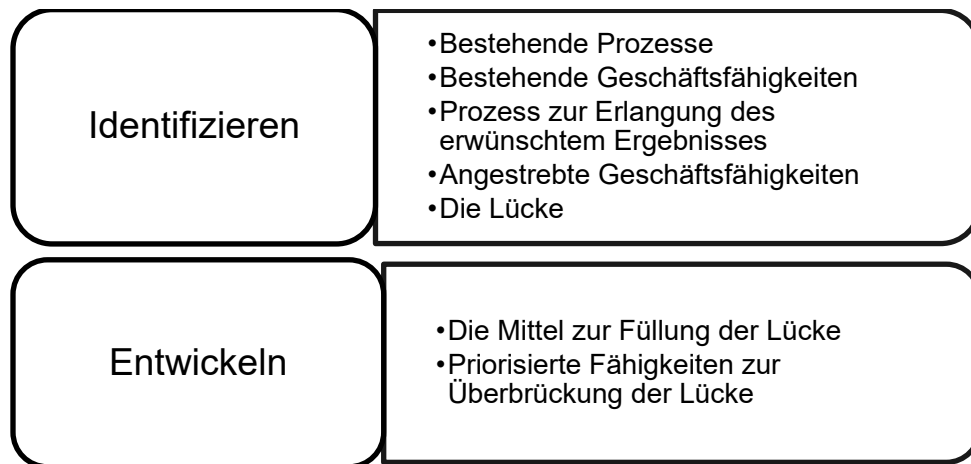


Abbildung 8-5: Der Vorgang der Lückenanalyse

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.

Beispiel:

VMFoods, eine Lebensmittelkette, möchte sich durch ein E-Commerce Portal im Internet erweitern und hat Ihre Firma mit der Erstellung des Produkts beauftragt.

Projektvision: Entwicklung eines einfach nutzbaren und ästhetisch ansprechenden Internet-Verkaufskanals für VMFoods.

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 Stakeholder Identifizieren

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

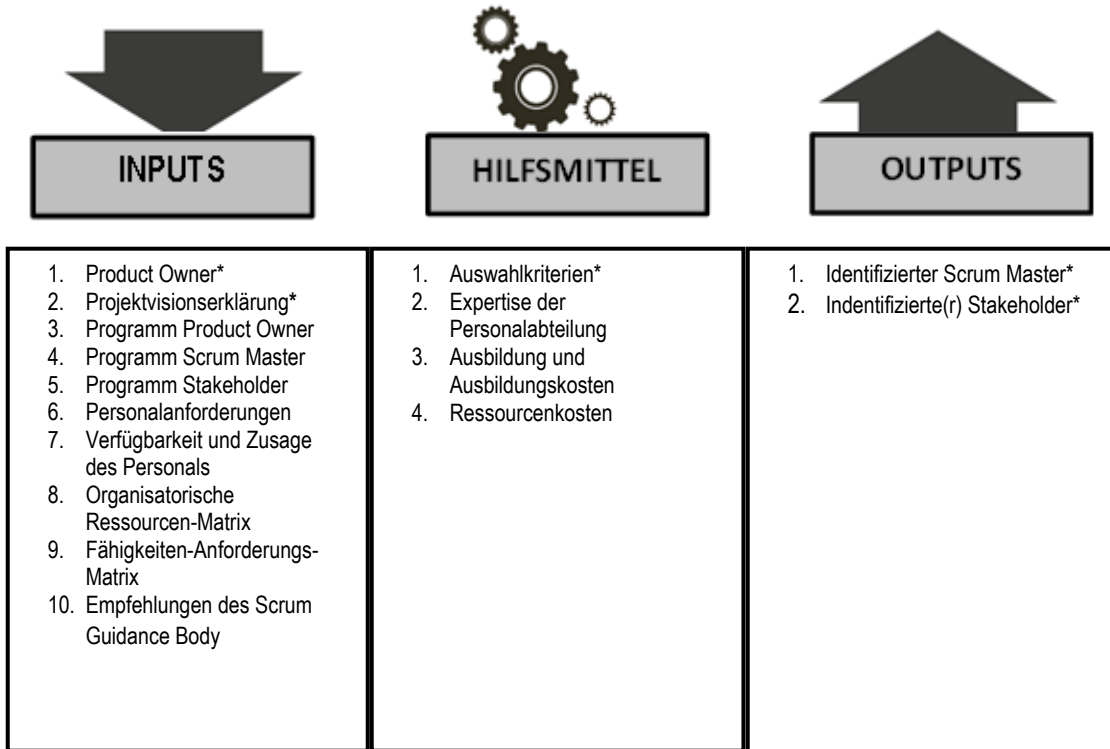


Abbildung 8-6: Scrum Master und Stakeholder identifizieren—Inputs, Hilfsmittel und Outputs

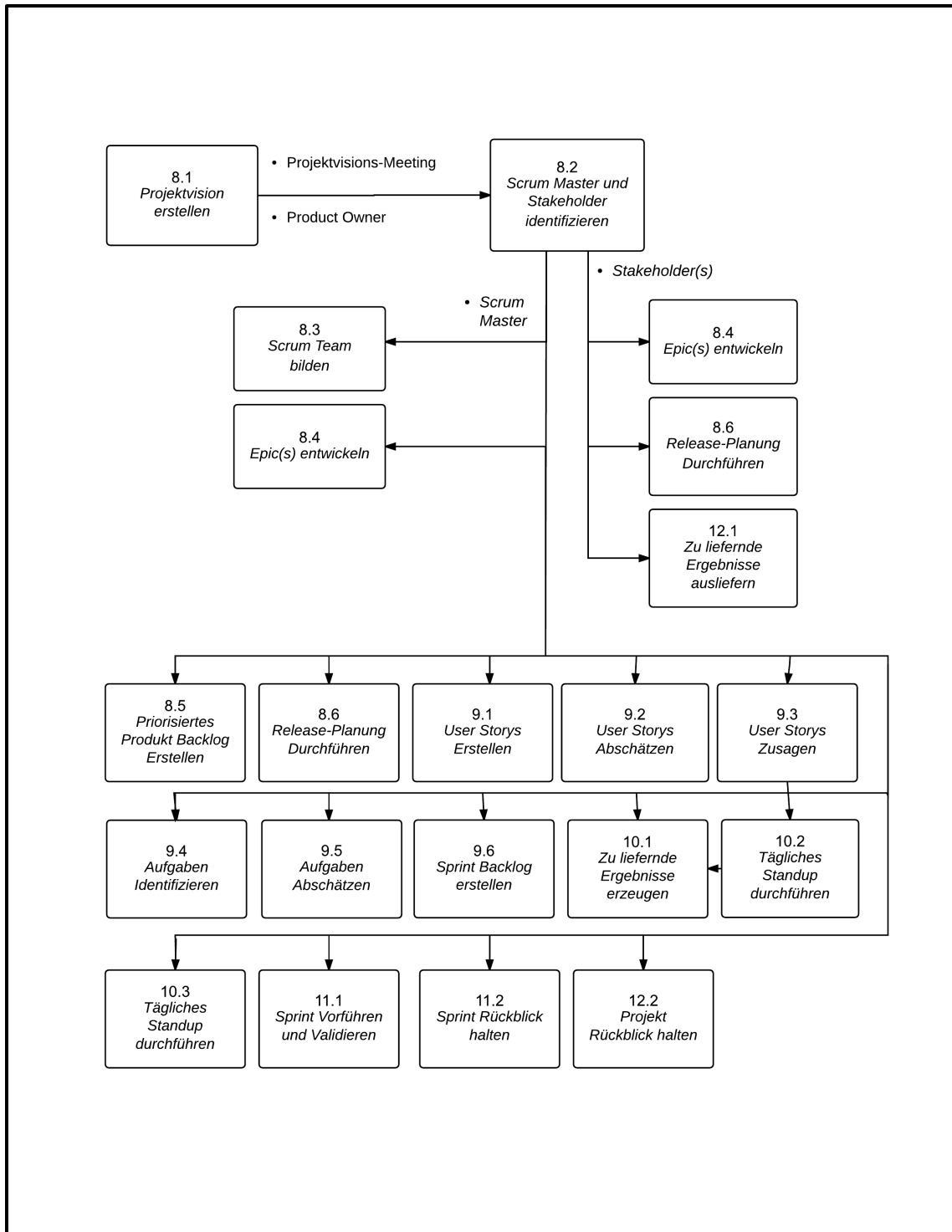


Abbildung 8-7: Scrum Master und Stakeholder Identifizieren—Datenflussdiagramm

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 Programm Product Owner

Siehe Abschnitt 3.7.3.1.

8.2.1.4 Programm Scrum Master

Siehe Abschnitt 3.7.3.2.

8.2.1.5 Programm Stakeholder

Siehe Abschnitt 8.1.1.4.

8.2.1.6 Personalanforderungen

Das Identifizieren der Personalanforderungen ist einer der ersten Schritte auf dem Weg zur Auswahl des Scrum Masters und der Stakeholder. Es ist wichtig, die Rollen und Verantwortungsbereiche von all jenen, welche an der Erfüllung der Aufgaben des Projekts beteiligt sind, zu dokumentieren. Dies umfasst jeden, der in irgendeiner Form an einem Projekt beteiligt ist – sei es in einer Kern- oder Nebenrolle.

Üblicherweise arbeiten der Product Owner oder der Scrum Master mit der Personalabteilung des Unternehmens zusammen, um die Personalanforderungen für das Projekt festzustellen und festzulegen.

8.2.1.7 Verfügbarkeit und Zusage des Personals

Vor der Auswahl des Scrum Masters und der Stakeholder 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.

8.2.1.8 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.

8.2.1.9 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.

8.2.1.10 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 Stakeholder 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. *Dienender Leader-Stil*—Abschnitt 3.10.4.1 bietet hierzu weitere Details

Beim Identifizieren der Stakeholder ist es wichtig zu beachten, dass Stakeholder 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 Stakeholder 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 Stakeholder 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 Stakeholder zu identifizieren.

8.2.2.3 **Ausbildung und Ausbildungskosten**

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 Stakeholder 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*

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) Stakeholder*

Stakeholder, 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 Stakeholder sind diejenigen, für die das Projekt gemeinschaftliche Vorteile generiert.

Die Rolle des Stakeholders wird in Abschnitt 3.3.2 beschrieben.

8.3 Scrum Team bilden

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

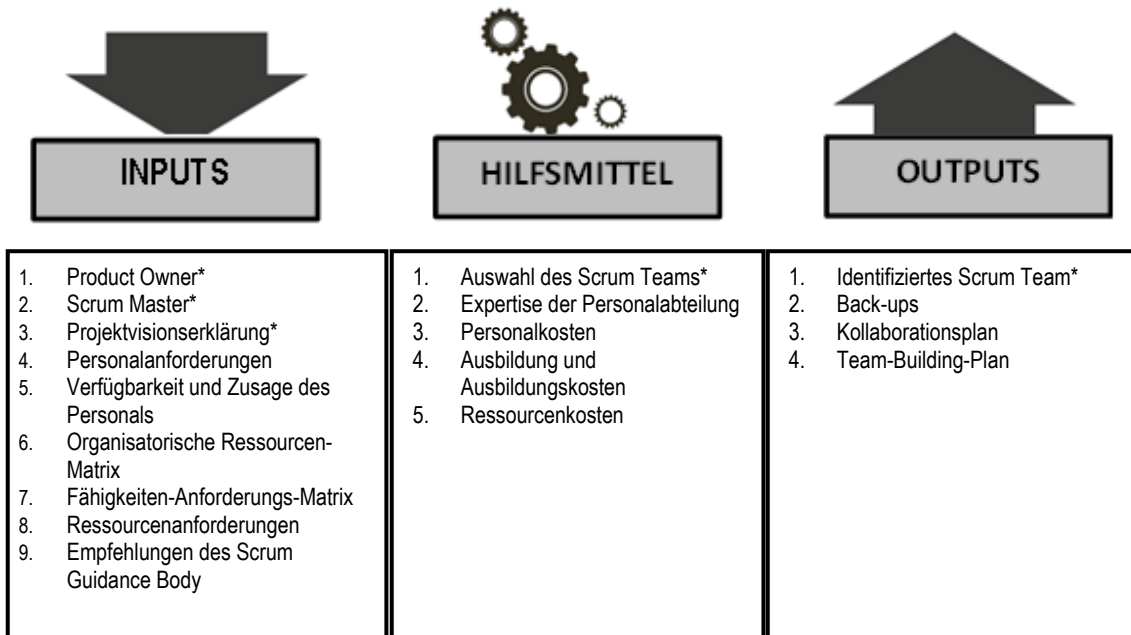


Abbildung 8-8: Scrum Team bilden–Inputs, Hilfsmittel und Outputs

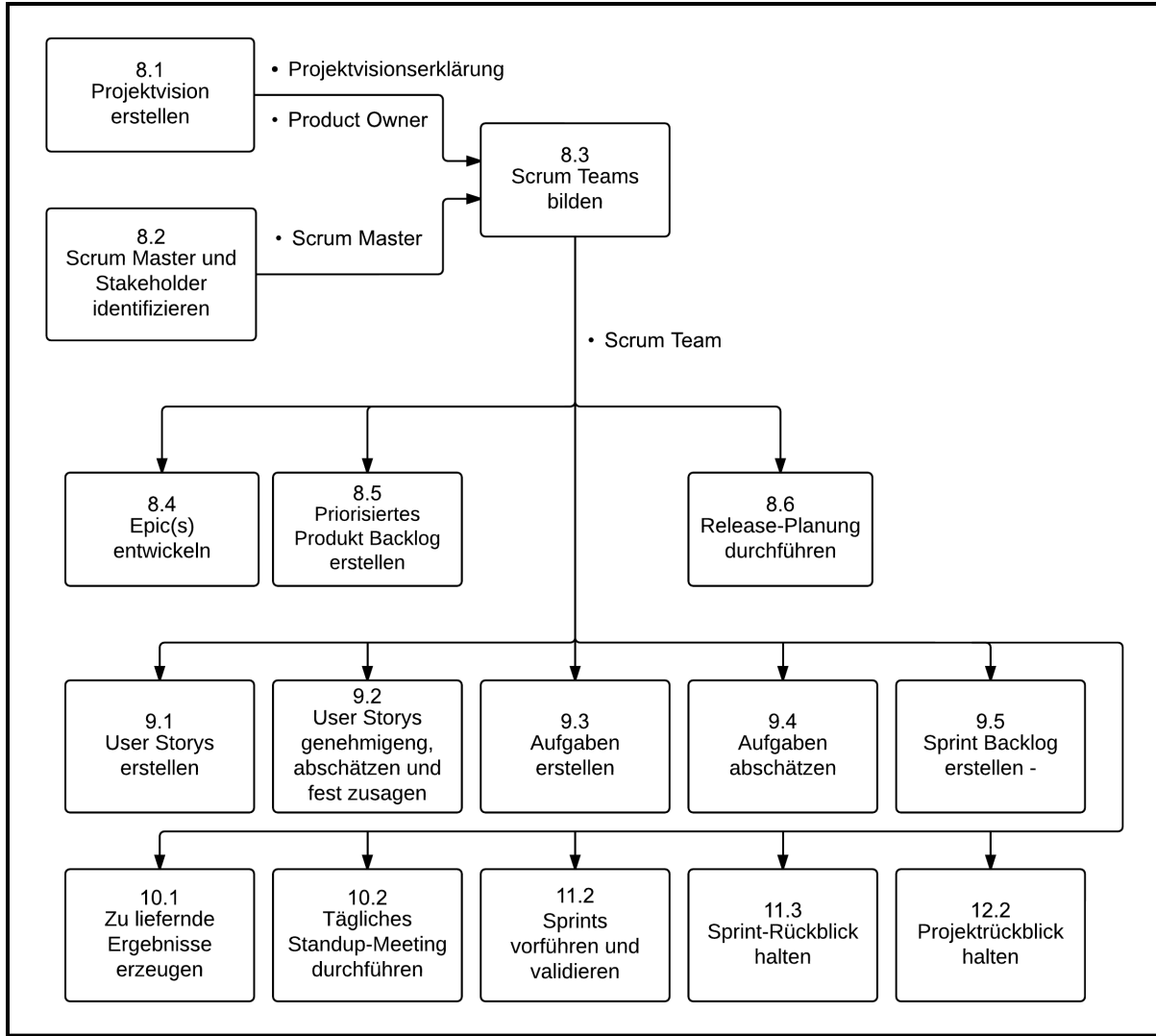


Abbildung 8-9: Scrum Team bilden–Datenflussdiagramm

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 Auswahl des Scrum Teams*

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 und Ausbildungskosten

Es kann sein, dass Teammitglieder nicht immer das benötigte Wissen oder die Fähigkeiten besitzen, um spezielle Aufgaben auszuführen. Der Product Owner sollte die Notwendigkeit von Weiterbildung der potentiellen Teammitglieder einschätzen und diese Weiterbildung ermöglichen, wenn Wissenslücken festgestellt 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 Stakeholder 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.

8.3.3 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 potentiell 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

8.3.3.2 Back-ups

Ein weiterer wichtiger Aspekt bei der Auswahl von Teams ist es, Backups für jedes Scrum Team-Mitglied festzulegen. 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, 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 Stakeholdern, da Scrum unnötige Dokumentation vermeidet. Allerdings kann es sein, dass für größere, komplexere Projekte, insbesondere bei verteilten Teams, eine formellere Vereinbarung notwendig wird.

Der Plan kann die Art und Weise behandeln, mit der die Scrum Kernteam-Mitglieder, Stakeholder 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

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

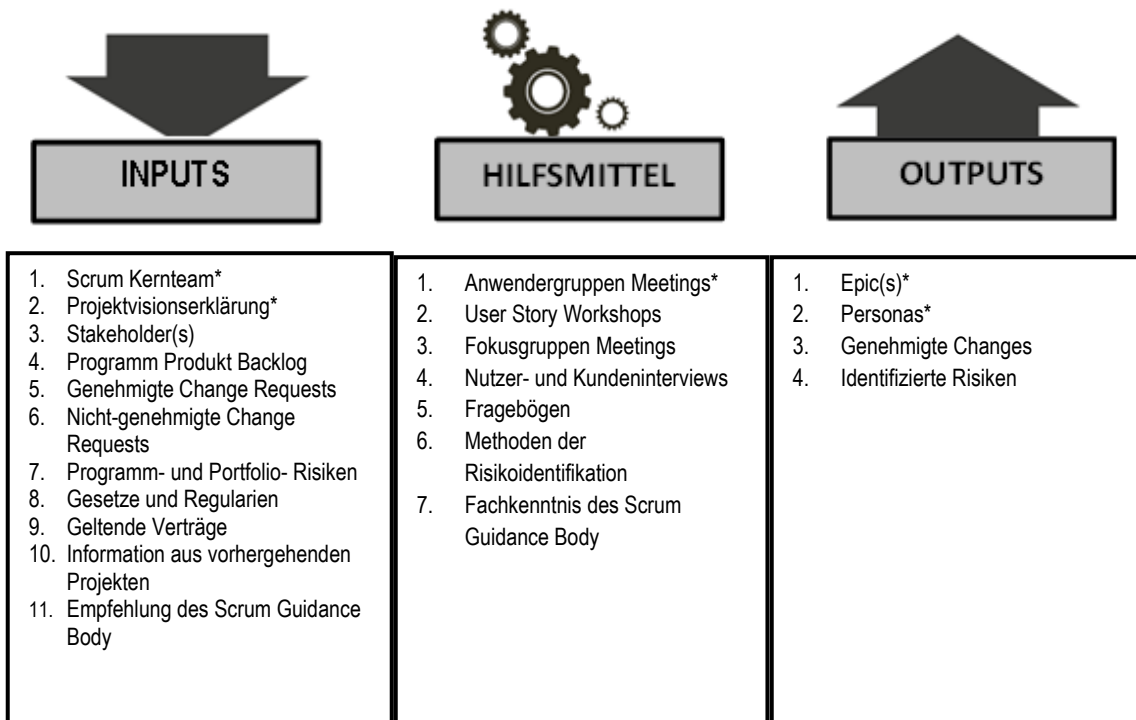


Abbildung 8-10: Epic(s) entwickeln–Inputs, Hilfsmittel und Outputs

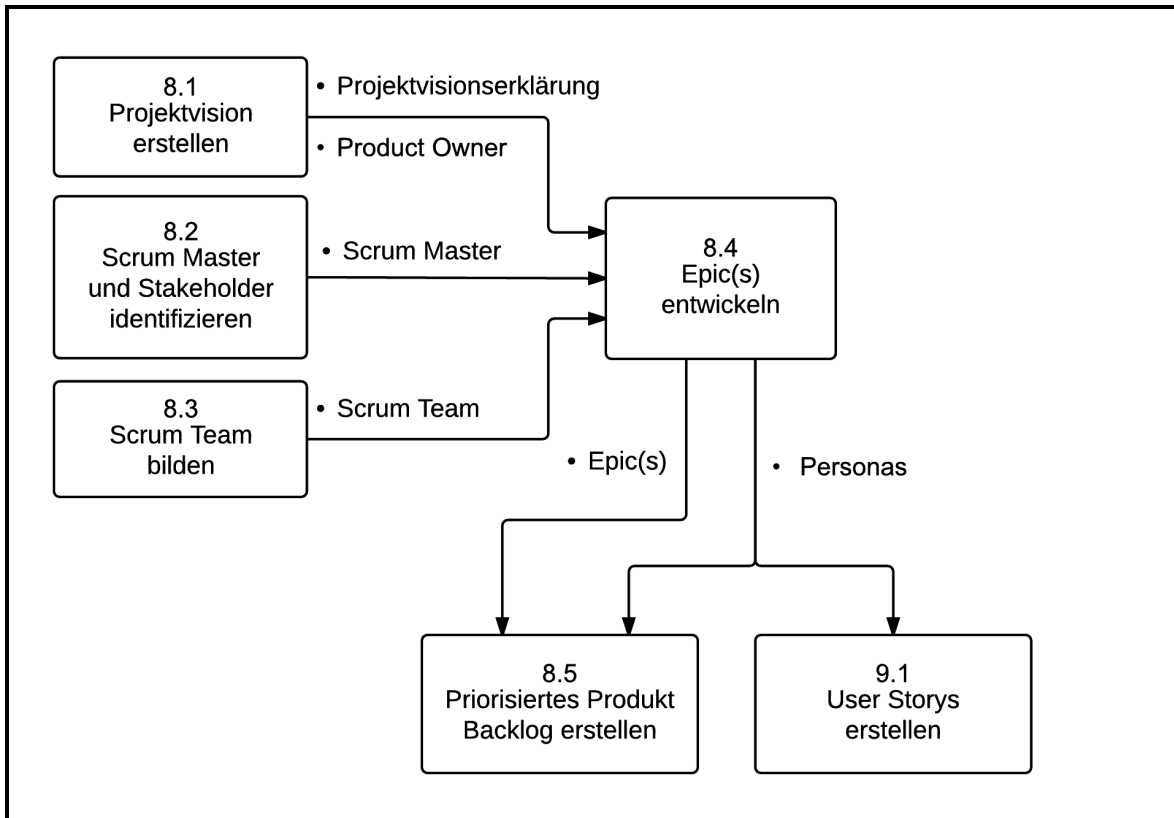


Abbildung 8-11: Epic(s) entwickeln-Datenflussdiagramm

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 Stakeholder

Siehe Abschnitt 8.2.3.2.

8.4.1.4 Programm Produkt Backlog

Siehe Abschnitt 8.1.1.6.

8.4.1.5 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.

8.4.1.6 Genehmigte Change Requests, 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.

8.4.1.7 Programm und Portfolio-Risiken

Risiken im Rahmen eines Portfolios oder Programms beeinflussen 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. Risiken von Programm und Portfolio können Vorgaben für den *Epic(s) entwickeln* Prozess sein und können einen Einfluss darauf haben, wie dieser Prozess gehandhabt wird.

Programm und Portfolio-Risiken werden im Abschnitt 7.5.1. beschrieben.

8.4.1.8 Gesetze und Regularien

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.

8.4.1.9 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 Inspektionspunkte in regelmäßigen Abständen. Dies hilft dem Kunden oder den Stakeholdern beim Treffen von Entscheidungen im Bezug auf Produktentwicklung, welche periodisch an jedem Inspektionspunkt 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.

8.4.1.10 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 vorherige Projekten können die Notizen des Projektmanagers, die Projektlogs, und Bemerkungen der Stakeholder beinhalten.

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

8.4.1.11 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 geschehen unter Beteiligung der relevanten Stakeholder (vorwiegend Anwender oder Kunden des Produkts) und versorgen das Scrum Kernteam mit Informationen aus erster Hand über die Erwartungen der Anwender. Dies hilft bei der Formulierung von Abnahmekriterien für das Produkt, und ermöglicht wertvolle Einsichten für die Entwicklung von Epic(s). Anwender-Gruppen Meetings sind für die Prävention kostenaufwändiger Nacharbeiten, welche aus Unklarheiten in Bezug auf Erwartungen und Anforderung entstehen können, von großer Bedeutung. Diese Meetings fördern auch die Akzeptanz des Projekts, und erzeugen ein gemeinsames Verständnis zwischen Scrum Kernteam und relevanten Stakeholder(n).

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 Stakeholder 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 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 Stakeholder
- Der Sammlung von Meinungen und Fakten
- Dem Verständnis für die Perspektive der Stakeholder hinsichtlich des Endprodukts
- Der Sammlung von Feedback über ein wiederholtes/teilweise entwickeltes Produkt.

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 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 Schlüsselparmeter 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.

8.4.3 Outputs

8.4.3.1 Epic(s)*

Epics werden in den frühen Phasen des Projekts niedergeschrieben, wenn die meisten User Storys high-level Funktionalitäten oder Produktbeschreibungen sind, und Anforderungen sehr allgemein definiert sind. Es sind große, nicht verfeinerte User Storys im Priorisierten Produkt Backlog.

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 Stakeholder, 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 Stakeholder. 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

Abbildung 8-12 zeigt um so Inputs, Hilfsmittel und Outputs für *Priorisiertes Produkt Backlog Erstellen* Prozess.

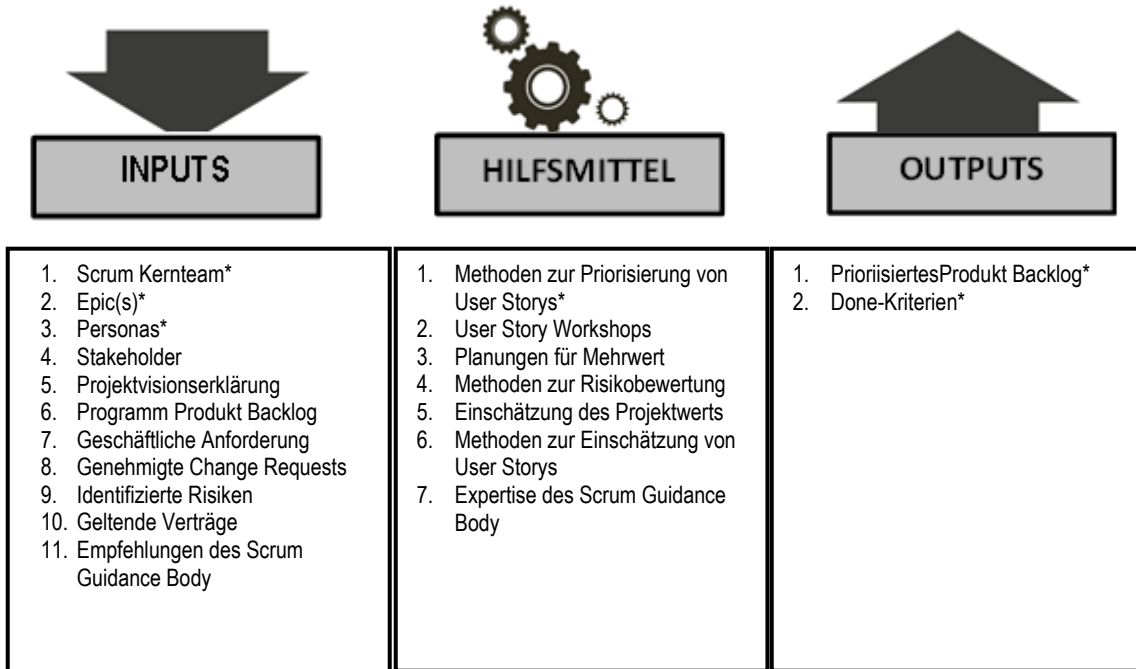


Abbildung 8-12: Priorisiertes Produkt Backlog Erstellen–Inputs, Hilfsmittel und Outputs

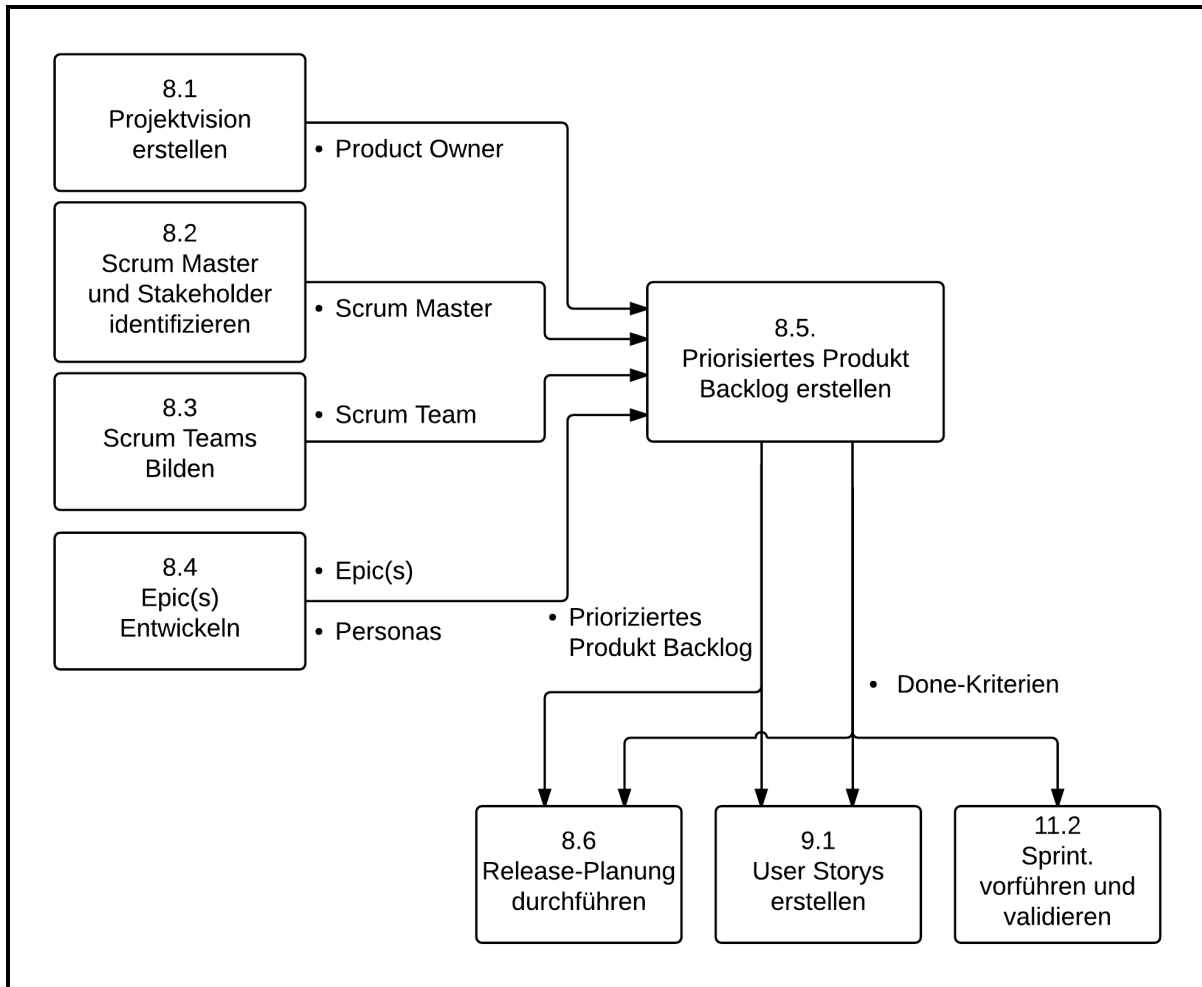


Abbildung 8-13: Erstellen eines priorisierten Produkt Backlogs: Datenflussdiagramm

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 Stakeholder

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 **Planung für Mehrwert**

Siehe Abschnitt 4.5.2.

8.5.2.4 **Methoden der Risikobewertung**

Siehe Abschnitt 7.4.2.1.

8.5.2.5 **Abschätzung des Projektwerts**

Siehe Abschnitt 4.5.1.

8.5.2.6 Abschätzungsmethoden für User Storys

Alle Hilfsmittel, welche für die Einschätzung von User Storys (vergleiche Abschnitt 9.2.2.3) genutzt werden, können auch für das Erzeugen von high-level Abschätzungen von Epics bei der Erstellung des priorisierten Produkt Backlogs angewendet werden. Solche wichtigen Hilfsmittel sind beispielsweise:

1. Wideband Delphi
2. Planungspoker
3. Faust von Fünf
4. Story Points zur Kostenabschätzung
5. Vergleichende Abschätzung

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.

8.5.3 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**—Es ist üblicherweise nicht möglich, ein priorisiertes Produkt Backlog zu erstellen, in dem keine Abhängigkeiten zwischen User Storys bestehen. Funktionale Anforderungen hängen oft von anderen funktionalen oder gar non-funktionalen Anforderungen ab. Diese Abhängigkeiten können beeinflussen, wie die User Storys in einem priorisierten Produkt Backlog priorisiert werden. Zwei der verbreitetsten Methoden, diese Abhängigkeiten zu lösen, sind das Aufteilen einer einzelnen Story in mehrere Teile oder das kombinieren von voneinander abhängigen Storys.
- **Abschätzungen**—High-Level Abschätzungen für Epic(s) sind ebenfalls im priorisierten Produkt Backlog vorhanden.

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.

8.6 Release-Planung Durchführen

Abbildung 8-14 zeigt Inputs, Hilfsmittel und Outputs für *Release-Planung Durchführen* Prozess.

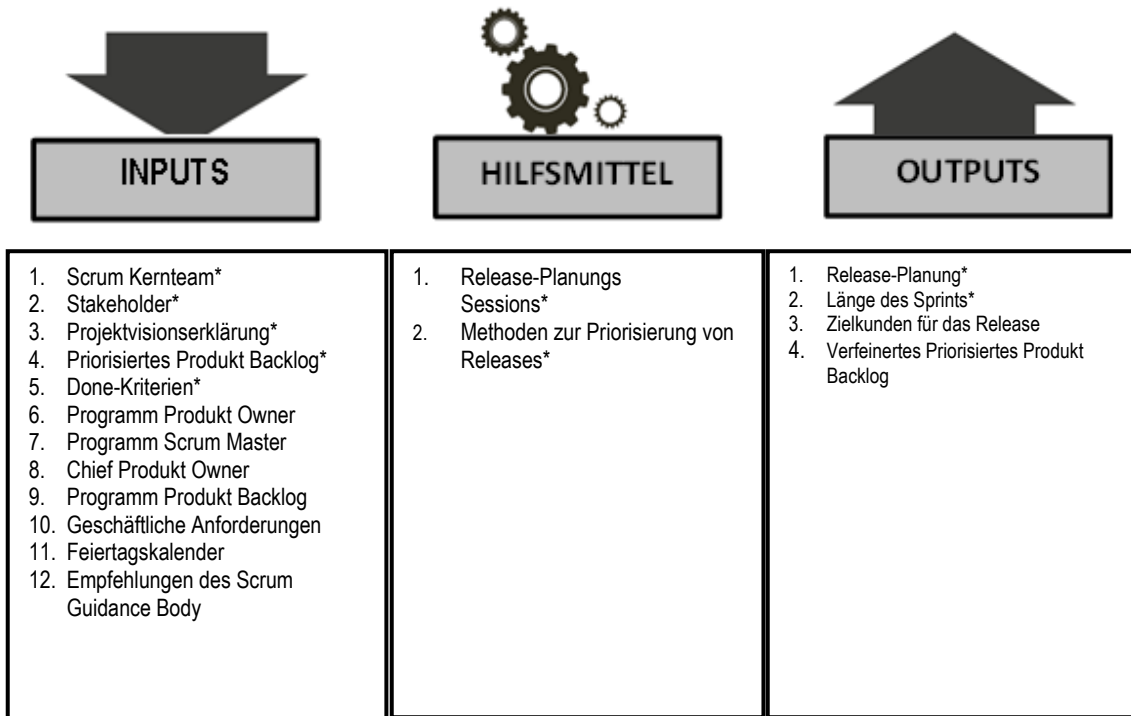


Abbildung 8-14: Release-Planung durchführen-Inputs, Hilfsmittel und Outputs

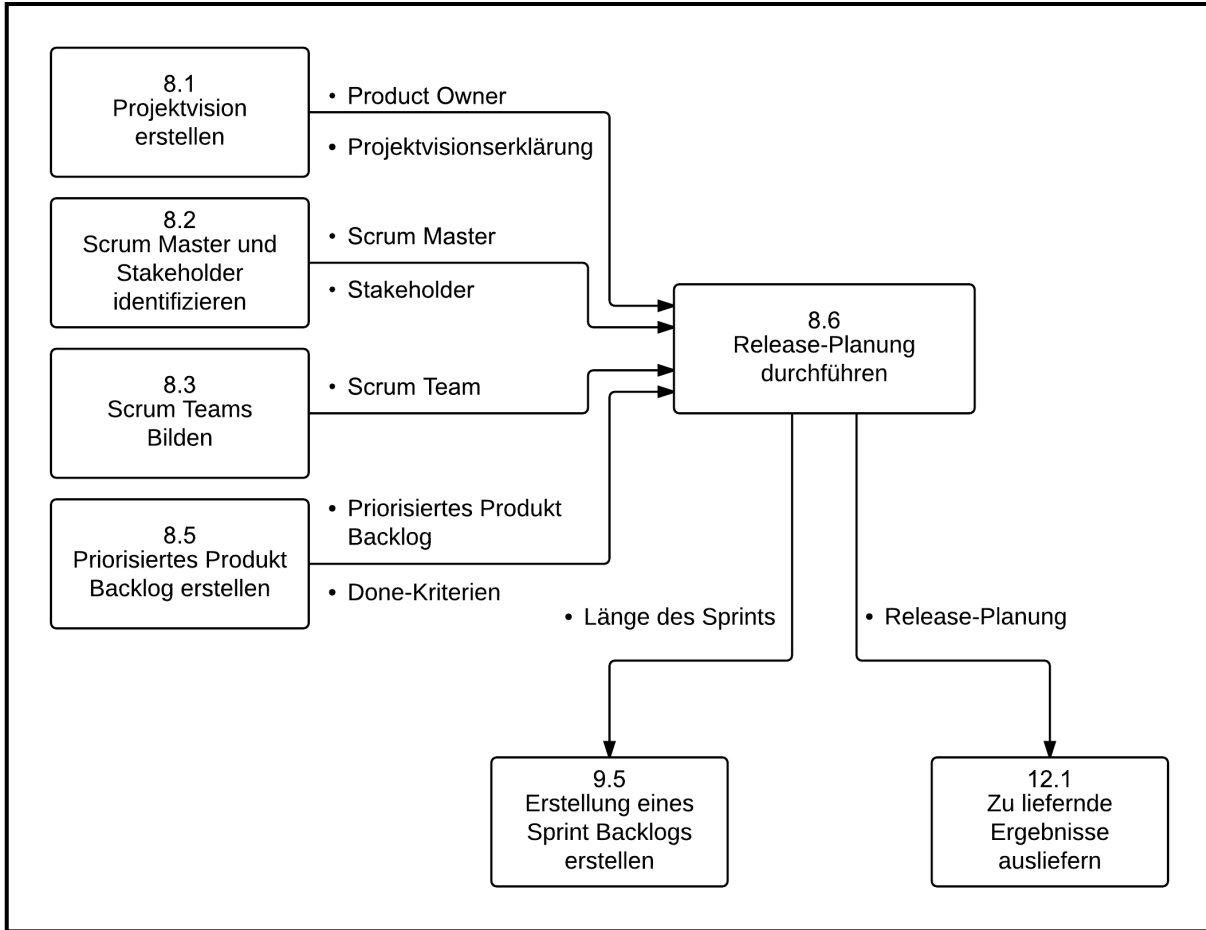


Abbildung 8-15: Release-Planung durchführen–Datenflussdiagramm

8.6.1 Inputs

8.6.1.1 Scrum Kernteam*

Siehe Abschnitt 8.4.1.1.

8.6.1.2 Stakeholder*

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 Programm Product Owner

Siehe Abschnitt 8.1.1.2.

8.6.1.7 Programm Scrum Master

Siehe Abschnitt 8.1.1.3.

8.6.1.8 Programm Produkt Backlog

Siehe Abschnitt 8.1.1.6.

8.6.1.9 Geschäftliche Anforderungen

Siehe Abschnitt 8.5.1.7.

8.6.1.10 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.

8.6.1.11 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11

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.

8.6.2 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 Stakeholder (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 datumsgeprägt, 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. Nicht nach jeden Sprintdurchlauf 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.

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.

Ein Sprint kann auf 1 bis 6 Wochen zeit-begrenzt werden. Um ein Maximum an Nutzen aus einem Scrum Projekt zu ziehen, ist es jedoch, immer ratsam die Länge des Sprints auf maximal 4 Wochen zu halten, wenn es nicht um Projekte mit sehr stabilen Anforderungen geht, bei denen die Sprintlänge auf bis zu 6 Wochen verlängert werden kann.

Die Auswirkungen einer zu erwartenden Änderung der Sprintlänge wird in Abschnitt 6.5.1 beschrieben.

8

8.6.3.3 Zielkunden für das Release

Nicht jedes Release richtet sich an alle Stakeholder oder Anwender. Die Stakeholder 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 Stakeholdern die Release-Planungs-Sitzungen abgehalten hat.

8.7 Datenflussdiagramm der Phase

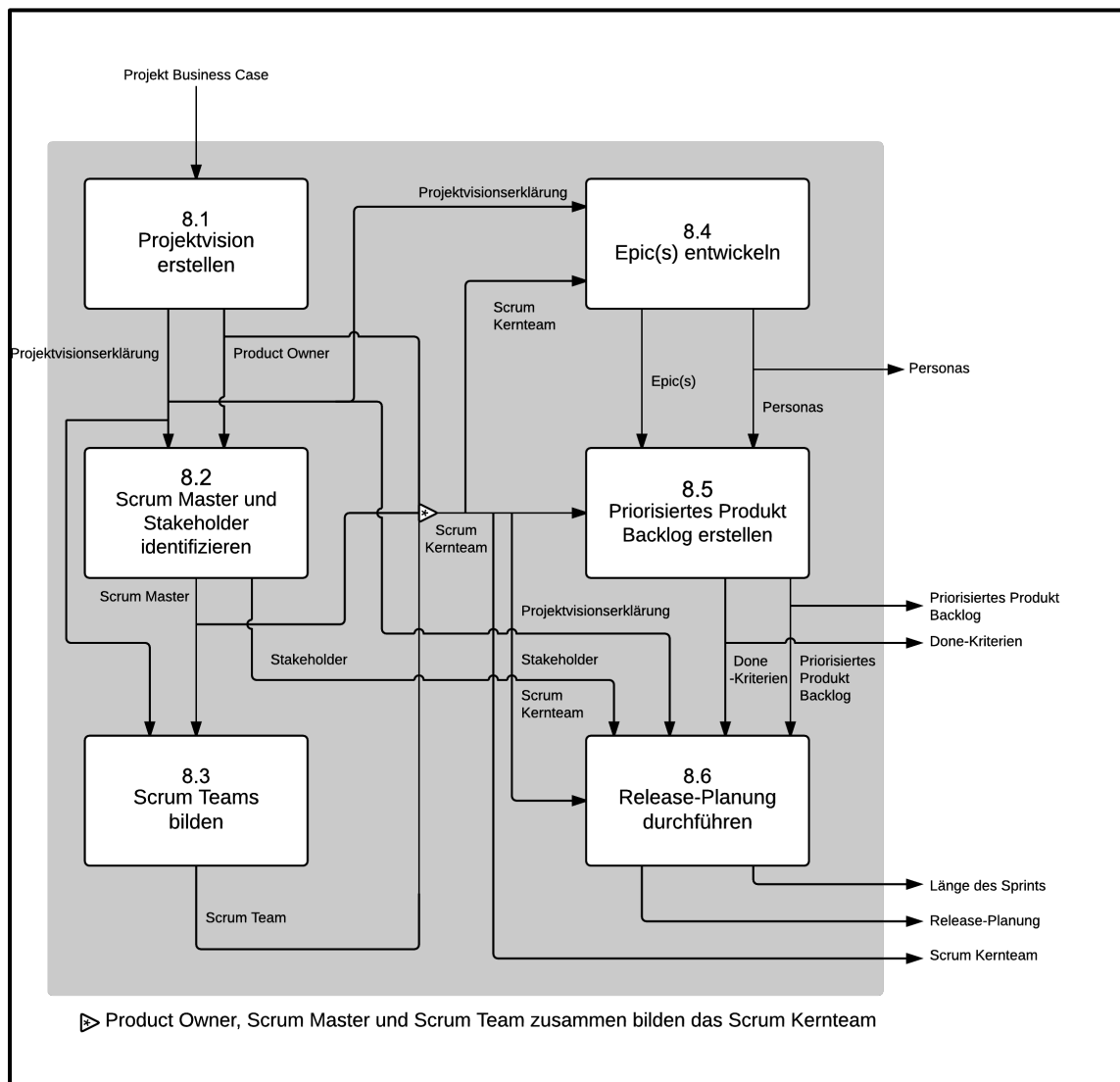


Abbildung 8-16: Initiierungsphase-Datenflussdiagramm

9. PLANEN UND ABSCHÄTZEN

Die Planungs- und Schätzungsphase 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 erstellen*.

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 Stakeholder 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.

Die Abbildung 9-1 zeigt eine Übersicht über die Prozesse der Planungs- und Schätzungsphase, welche sich folgendermaßen aufteilen:

9.1 User Storys Erstellen — Während dieses Prozesses werden User Storys und ihre zugehörigen Abnahmekriterien erstellt. User Storys werden normalerweise vom Product Owner geschrieben. Sie sind so konzipiert, dass sichergestellt werden kann, dass die Bedürfnisse des Kunden klar dargestellt und von allen

Stakeholdern komplett verstanden werden können. User Story Schreib-Workshops können durchgeführt werden, bei denen die Mitglieder des Scrum Teams die User Storys erstellen. User Storys werden in das Priorisierte Produkt Backlog eingearbeitet.

9.2 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.

9.3 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.

9.4 Aufgaben identifizieren — In diesem Prozess werden die zugesagten User Storys auf spezifische Aufgaben heruntergebrochen und in Form einer Aufgabenliste zusammengestellt.

9.5 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.

9.6 Sprint Backlog erstellen — Während dieses Prozesses hält das Scrum-Kernteam Sprint Planungs Meetings ab, in welchem die Gruppe ein Sprint Backlog erstellt. Dieses beinhaltet alle Aufgaben, die im Sprint erledigt werden müssen.

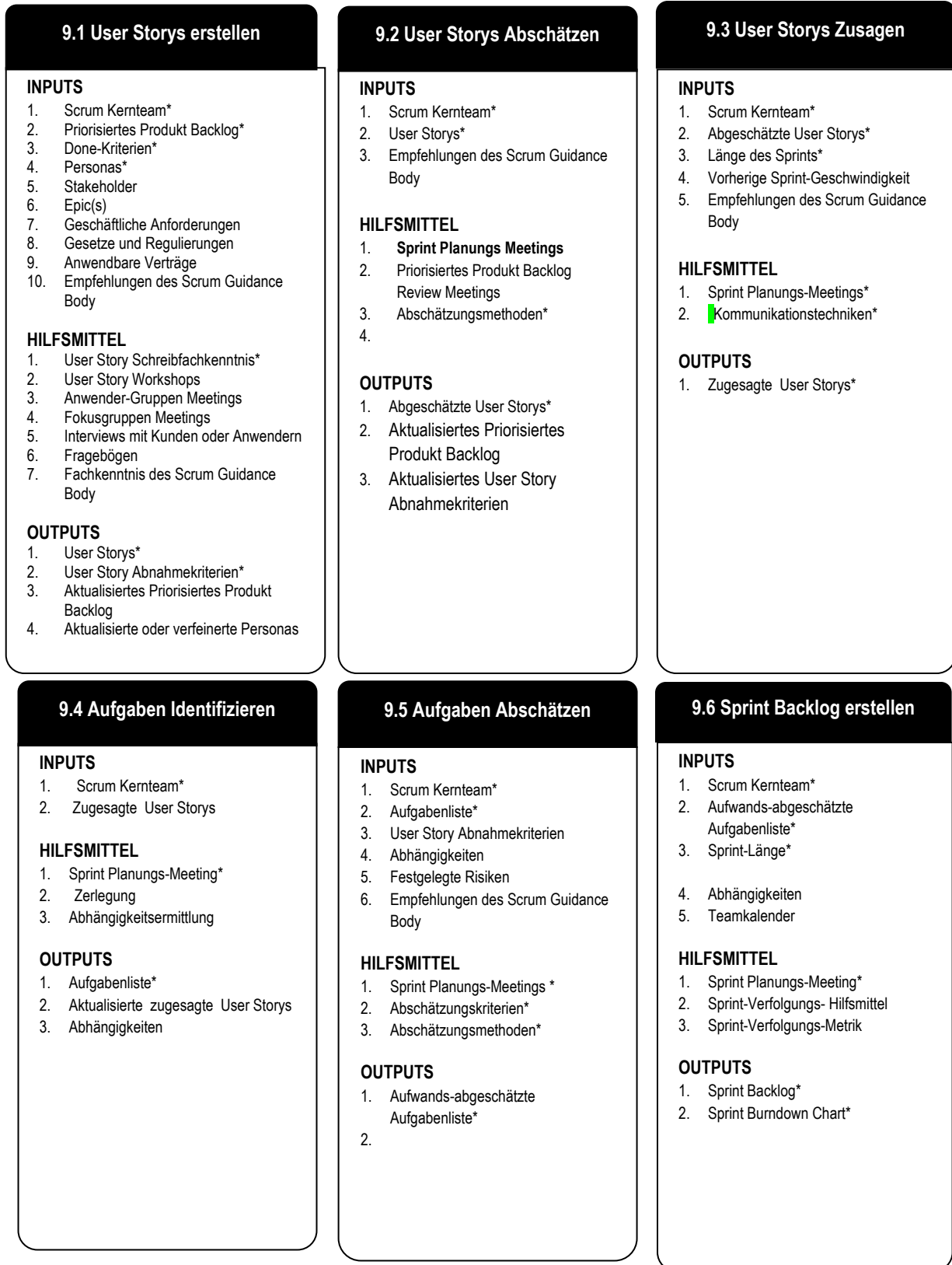


Abbildung 9-1: Planungs- und Abschätzungsübersicht

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

Abbildung 9-2 unten zeigt die verbindlichen Inputs, Hilfsmittel und Outputs für Prozesse in Planungs- und Schätzungsphase.

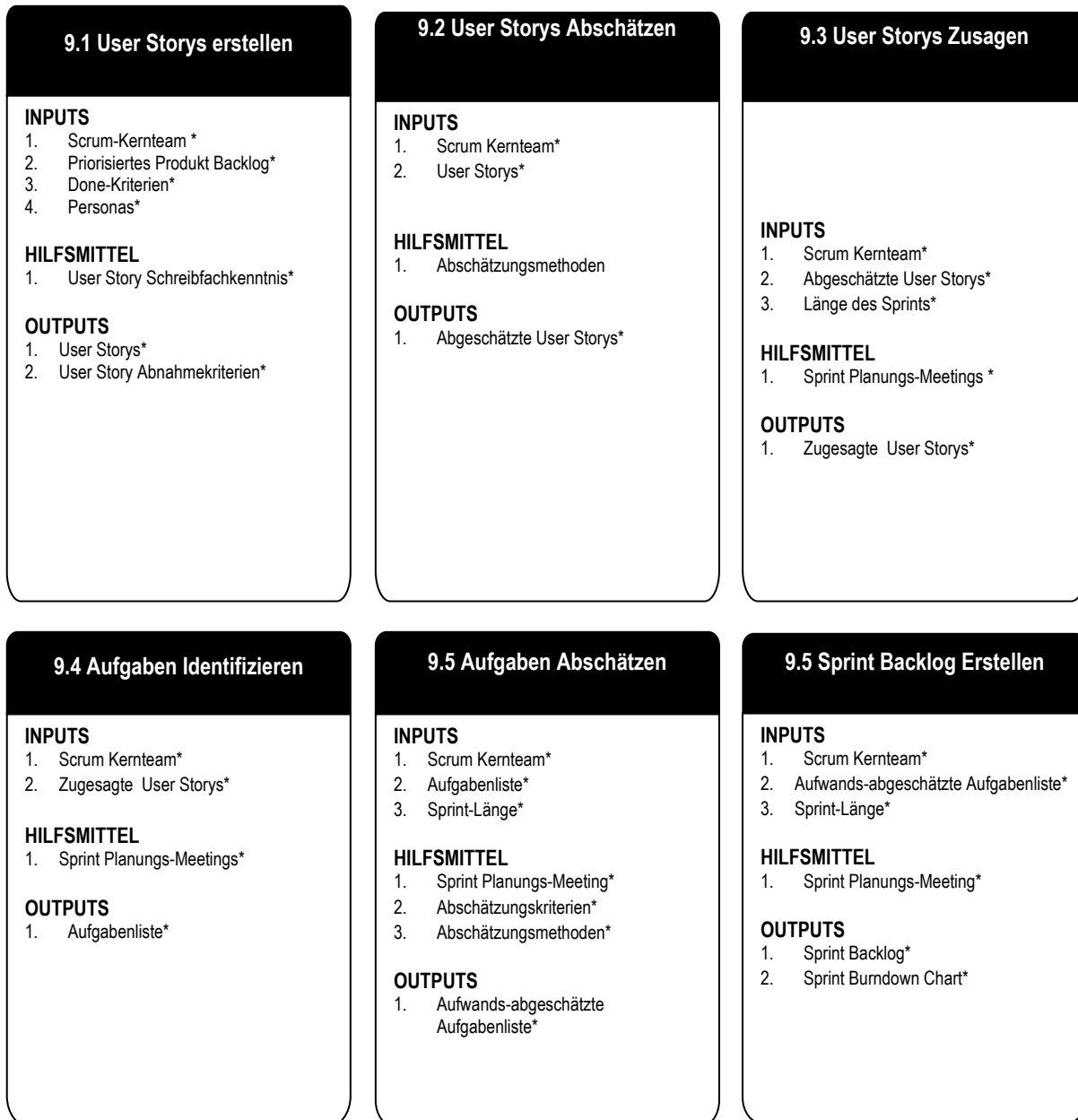


Abbildung 9-2: Planungs- und Abschätzungsübersicht (Kernpunkte)

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

9.1 User Storys erstellen

Abbildung 9-3 zeigt Inputs, Hilfsmittel und Outputs für den *User Storys Erstellen* Prozess.

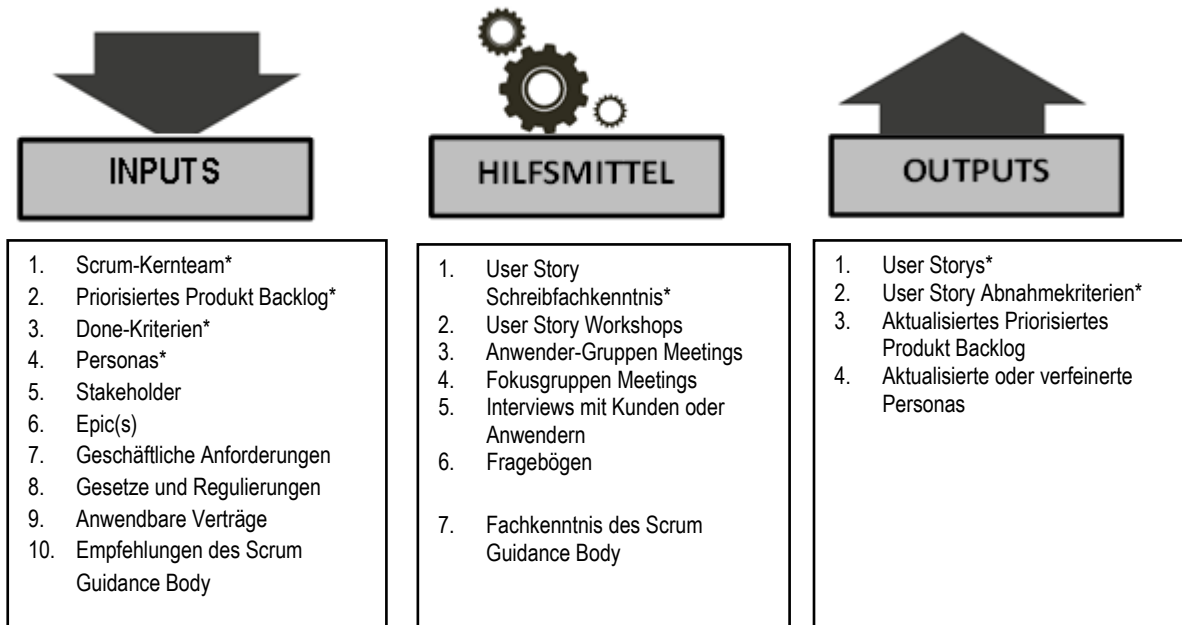


Abbildung 9-3: User Storys erstellen–Inputs, Hilfsmittel und Outputs

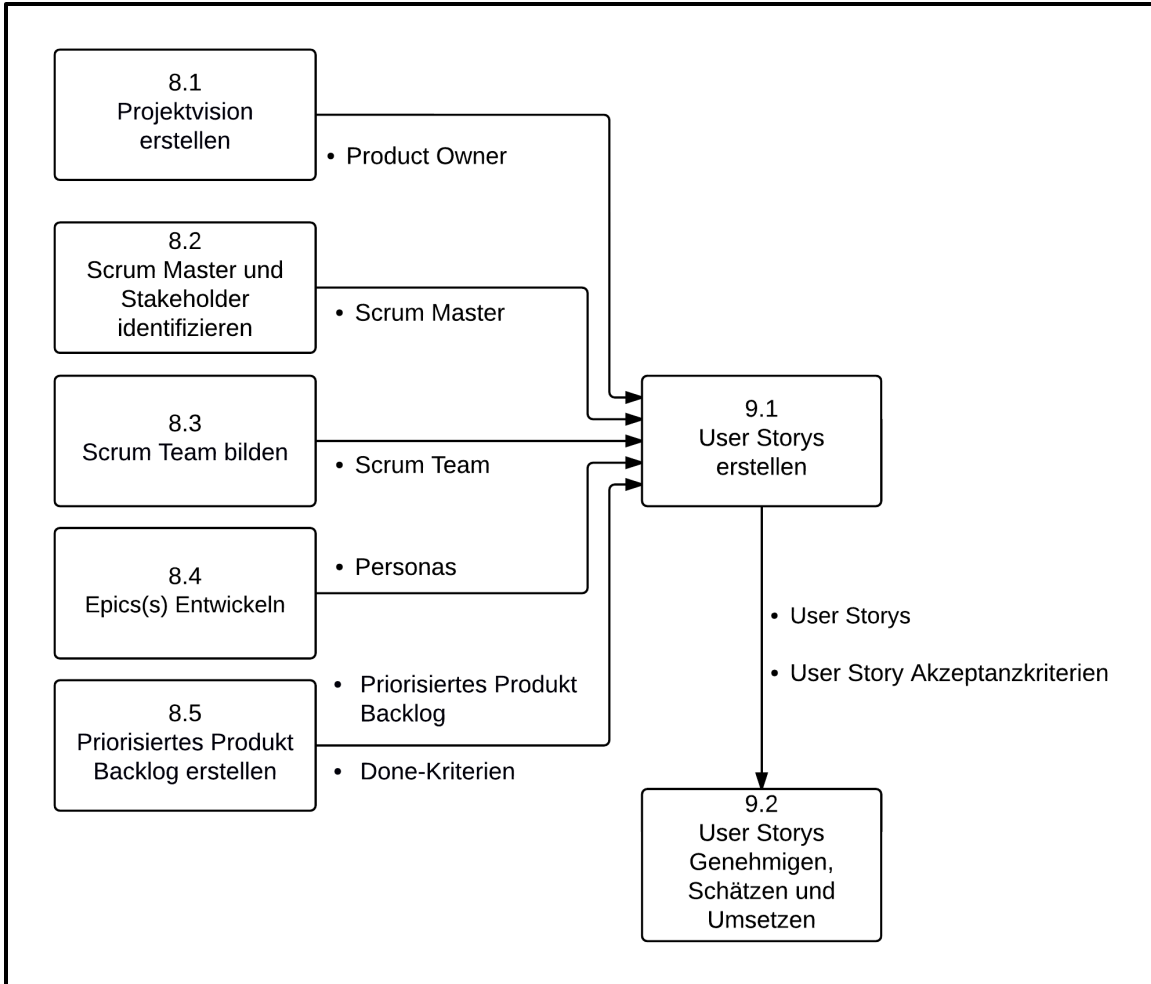


Abbildung 9-4: User Storys erstellen—Datenflussdiagramm

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 Stakeholder

Siehe Abschnitt 8.2.3.2.

9.1.1.6 Epic(s)

Siehe Abschnitt 8.4.3.1.

9.1.1.7 Geschäftliche Anforderungen

Siehe Abschnitt 8.5.1.7.

9.1.1.8 Gesetze und Regulierungen

Siehe Abschnitt 8.4.1.8.

9.1.1.9 Anwendbare Verträge

Siehe Abschnitt 8.4.1.9.

9.1.1.10 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 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

Fokusgruppen Meetings 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. Fokusgruppen Meetings folgen normalerweise einem bestimmten Format, bei welchem der Gruppe Fragen gestellt werden, die dann innerhalb der Gruppe diskutiert werden. Jedes Fokusgruppen Meeting 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 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 einige 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

Abbildung 9-5 zeigt um so Inputs, Hilfsmittel und Outputs für *User Storys Abschätzen* Prozess.

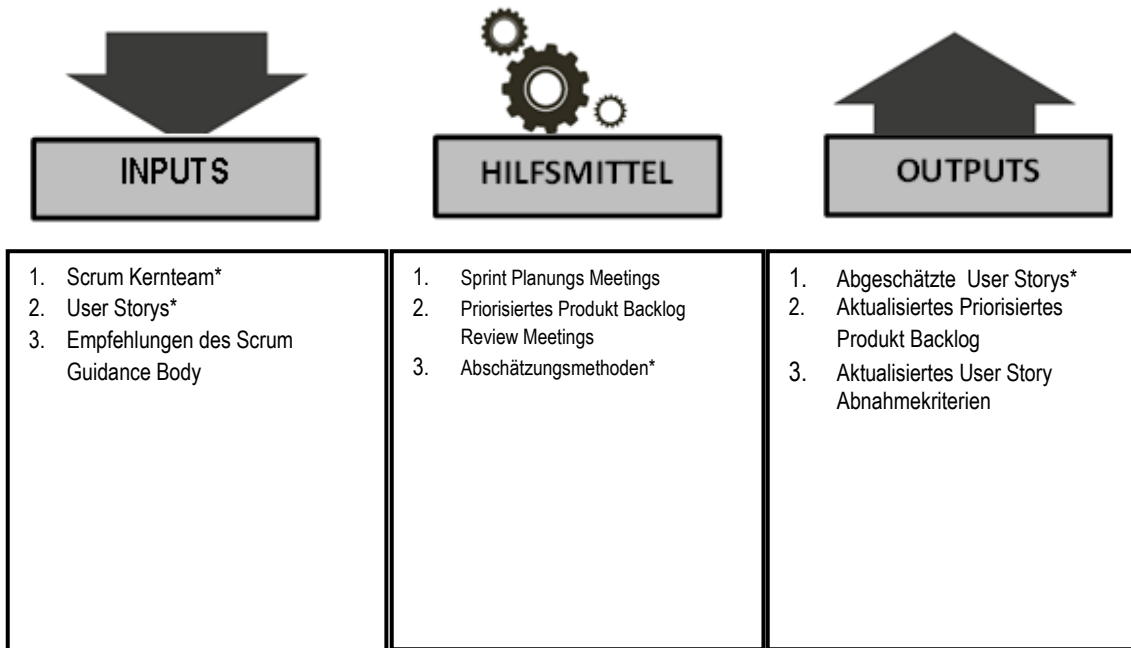


Abbildung 9-5: User Storys Genehmigen, Abschätzen und Zusagen–Inputs, Hilfsmittel und Outputs

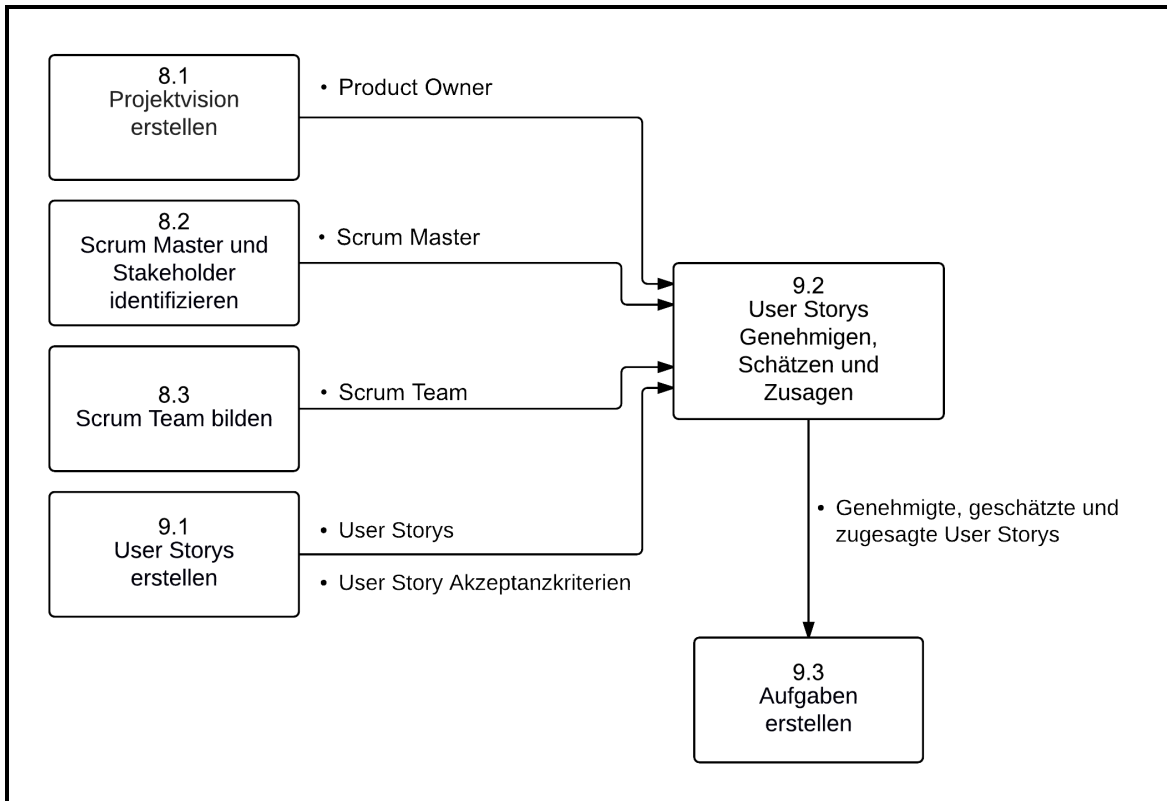


Abbildung 9-6: User Storys Genehmigen, Abschätzen und Zusagen – Datenflussdiagramm

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

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 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.

9.2.2 Hilfsmittel

9.2.2.1 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.

9.2.2.2 Priorisiertes Produkt Backlog Review Meetings

Priorisiertes Produkt Backlog Review Meetings werden als Teil des *Priorisiertes Produkt Backlog Pflegen* Prozesses durchgeführt. Wenn neue oder aktualisierte User Storys im Backlog verfeinert werden, führt das Scrum Team eine neue oder aktualisierte Abschätzung für jede User Story durch. Relativer Größe, oder Story Points, können für das Abschätzen der Gesamtgröße einer User Story oder eines Features verwendet werden. Dieser Ansatz weist einen Story Point Wert zu, der auf einer ganzheitlichen Bewertung der Größe beruht, wobei Risiko, Aufwand und Komplexität einbezogen werden. Diese Beurteilung wird vom Scrum

Team durchgeführt, und es wird ein Story Point Wert zugewiesen. Sobald die Bewertung einer User Story im Priorisierten Produkt Backlog abgeschlossen ist, kann das Scrum Team andere User Storys relativ zu dieser ersten User Story abschätzen.

9.2.2.3 Abschätzungsmethoden

Es gibt verschiedene Methoden, die zur Abschätzung von User Storys verwendet werden können. Einige wichtige Hilfsmittel 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 seine 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) 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.

1.

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.

Relative Größe, oder Story Points, können für das Abschätzen der Gesamtgröße einer User Story oder eines Features verwendet werden. Dieser Ansatz weist einen Story Point Wert zu, der auf einer ganzheitlichen Bewertung der Größe beruht, wobei Risiko, Aufwand und Komplexität einbezogen werden. Diese Beurteilung wird vom Scrum Team durchgeführt, und es wird ein Story Point Wert zugewiesen. Sobald die Bewertung einer User Story im Priorisierten Produkt Backlog abgeschlossen ist, kann das Scrum Team andere User Storys relativ zu dieser ersten User Story abschätzen.

Es sollte angemerkt werden, dass die die Anzahl an Story Points, die pro Sprint geliefert werden, nicht zum Vergleich von Teams herangezogen werden kann, da die Kalibrierung zwischen verschiedenen Teams oft sehr unterschiedlich ist.

9.3 User Storys Zusagen

Abbildung 9-7 zeigt Inputs, Hilfsmittel und Outputs für den *User Storys Zusagen* Prozess.

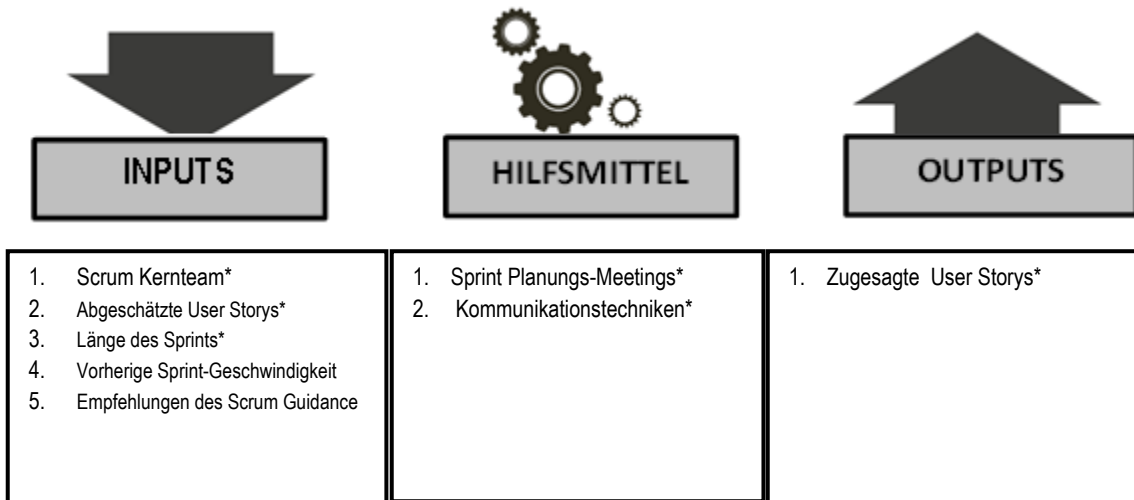


Abbildung 9-7: User Storys Zusagen-Inputs, Hilfsmittel und Outputs

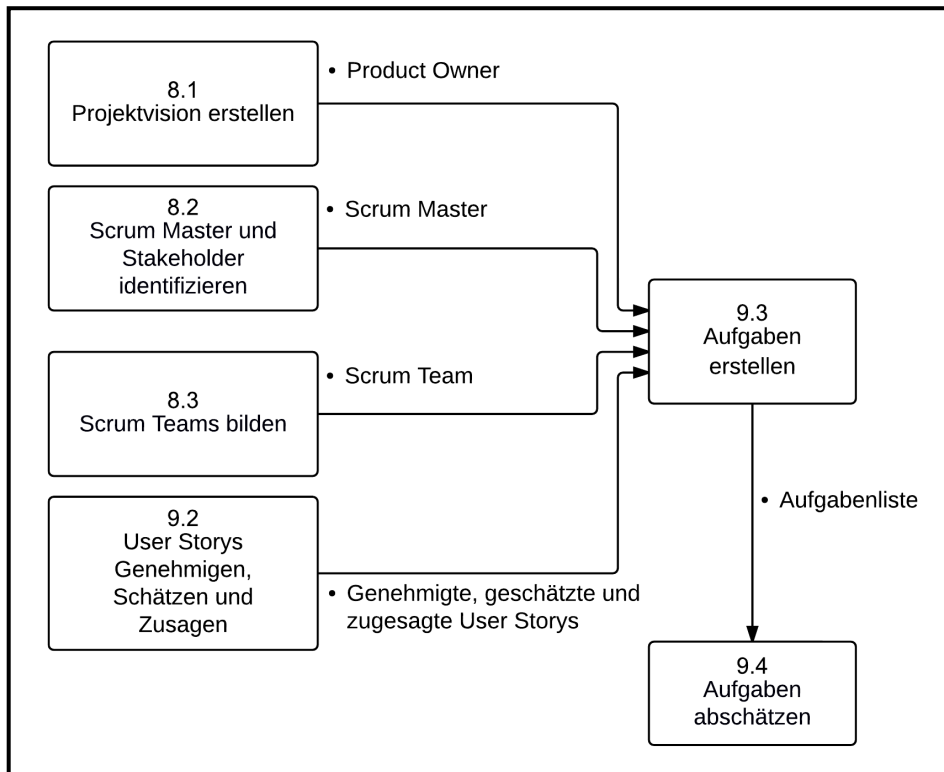


Abbildung 9-8: User Storys Zusagen-Datenflussdiagramm

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

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 Aufgabenplanungs-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 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.

9.3.2.3

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.

9.4 Aufgaben Identifizieren

Abbildung 9-9 zeigt Inputs, Hilfsmittel und Outputs für den *Aufgaben Identifizieren* Prozess.

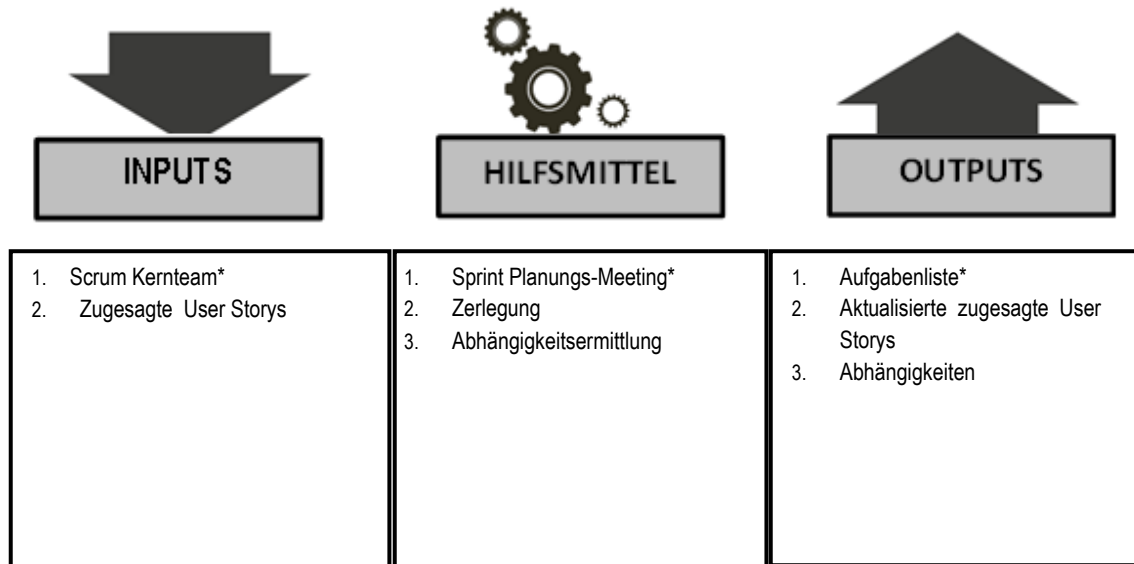


Abbildung 9-9: Aufgaben abschätzen–Inputs,Hilfsmittel und Outputs

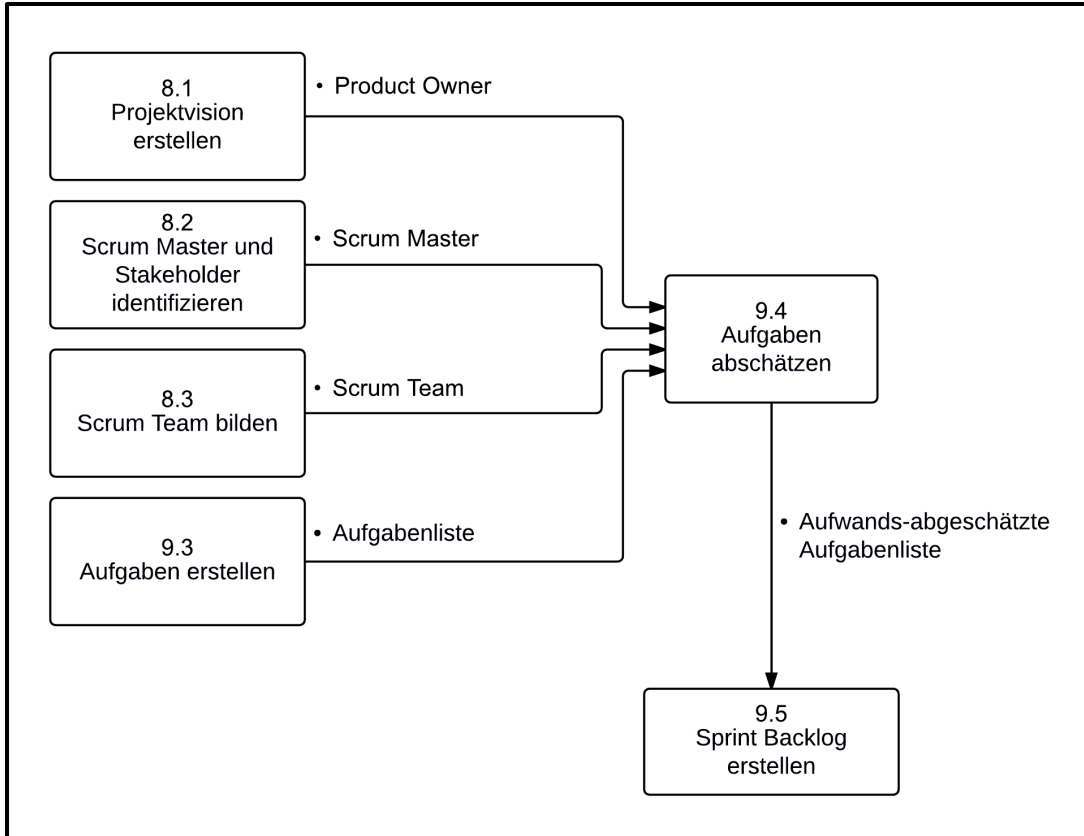


Abbildung 9-10: Aufgabenschätzung–Datenflussdiagramm

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

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.

9.4.2 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

Zerlegung ist ein Hilfsmittel, wodurch high-level Aufgaben in low-level und detailliertere Aufgaben aufgespalten werden. Die User Storys werden durch die Mitglieder des Scrum Teams in Aufgaben zerlegt. User Storys aus dem Priorisierten Produkt Backlog 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 9.3.2.4 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.

9.4.3 Outputs

9.4.3.1 9.3.3.1 Aufgabenliste *

Hierbei handelt es sich um eine umfassende Liste, die alle Aufgaben enthält, zu denen sich das Scrum Team im aktuellen Sprint verpflichtet hat. 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 9.3.3.2 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.

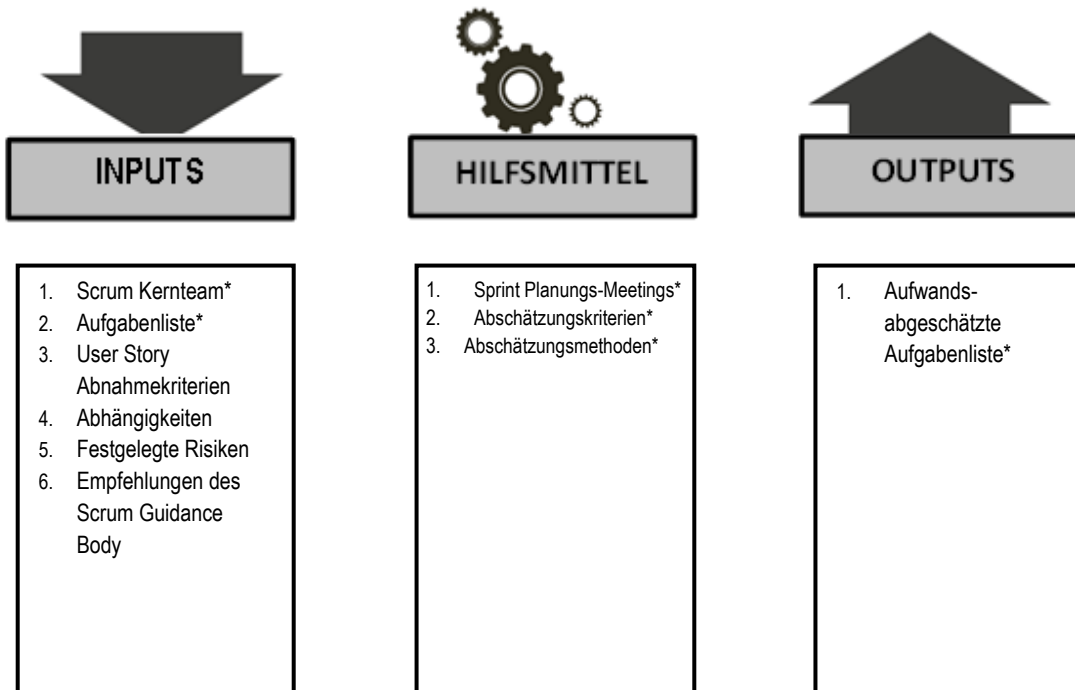
9.4.3.3 Abhängigkeiten

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

Abbildung 9-10 zeigt Inputs, Hilfsmittel und Outputs für den *Aufgaben Abschätzen* Prozess.



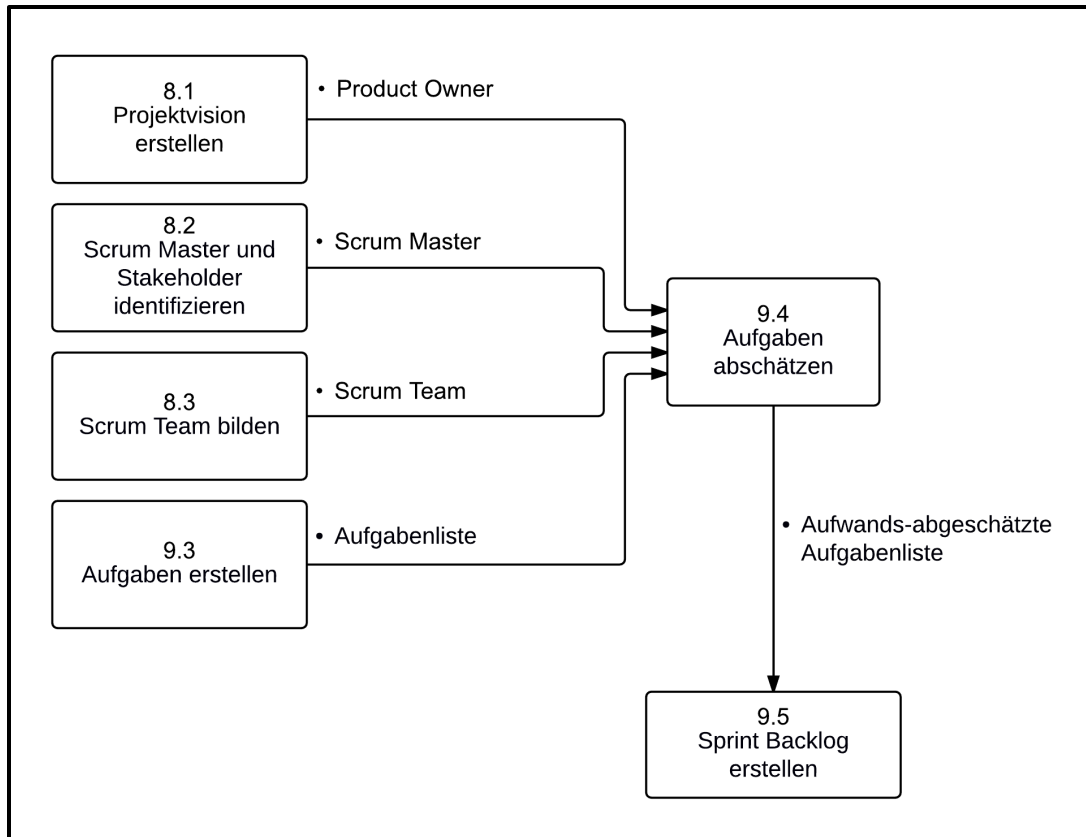


Abbildung 9-11: Aufgabenschätzung-Datenflussdiagramm

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

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 Festgelegte 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.

9.5.2 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. Während der Aufgaben Abschätzungs-Meetings benutzen die Mitglieder des Scrum 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

9.5.3 Outputs

9.5.3.1 Aufwands-abgeschätzte Aufgabenliste*

Die Aufwands-abgeschätzte Aufgabenliste ist eine Liste der in einem Sprint enthaltenen zugesagten User Storys. Typischerweise variiert die Genauigkeit der Schätzungen je nach Fähigkeiten des Teams. Die Aufwands-abgeschätzte Aufgabenliste wird vom Scrum Team während der Sprint Planungs Meetings zur Erstellung des Sprint Backlogs und des Sprint Burndown Charts verwendet.

9.5.3.2 Aktualisierte Aufgabenliste

Die Aufgabenliste, welche als Teil des Prozesses *Aufgaben erstellen* hervorgeht, beinhaltet die anfänglichen Schätzungen der User Storys, welche basierend auf den detaillierteren Schätzungsaktivitäten aus dem Prozess *Aufgaben Abschätzen angepasst werden müssen*. Es kann ebenfalls erneute Schätzungen, aufgrund eines Reviews von frühen Sprints oder eine Veränderung des gemeinschaftlichen Verständnisses der User Storys und Anforderungen, geben.

9.6 Sprint Backlog erstellen

Abbildung 9-12 zeigt Inputs, Hilfsmittel und Outputs für den *Sprint Backlog erstellen* Prozess.

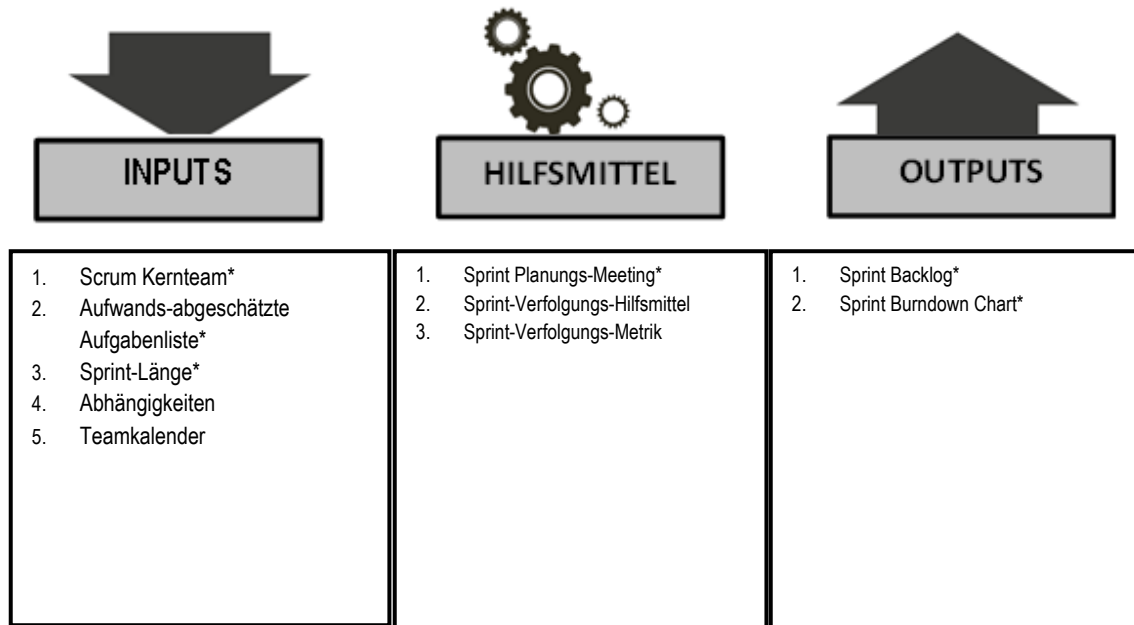


Abbildung 9-12: Sprint Backlog erstellen–Inputs, Hilfsmittel und Outputs

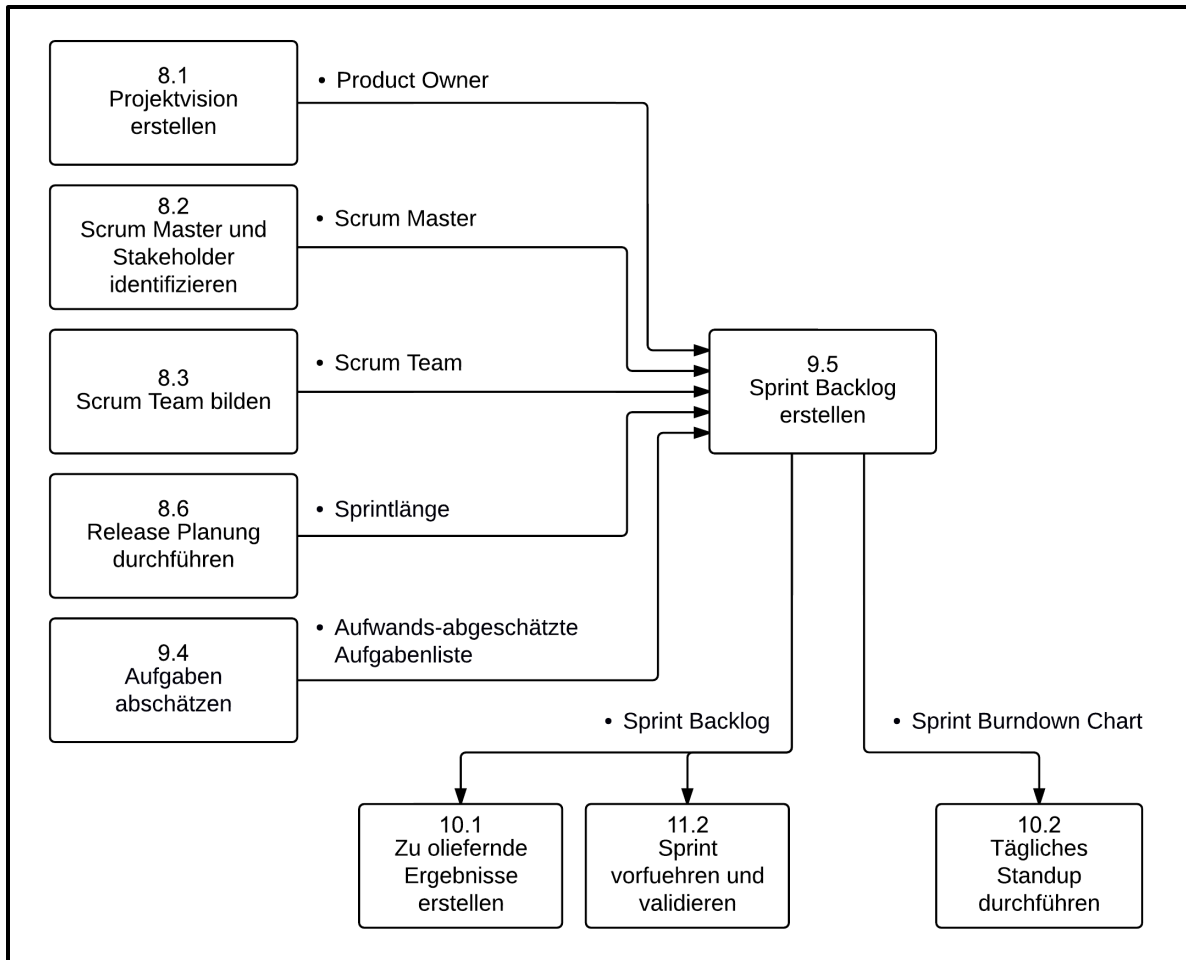


Abbildung 9-13: Sprint Backlog erstellen–Datenflussdiagramm

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: Unerledigt (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 gewichtetes Zählen genutzt werden.

9.6.3 **Outputs**

9.6.3.1 **Sprint Backlog***

Die Aufgabenliste, welche das Scrum Team im nächsten Sprint abarbeiten soll, wird als Sprint Backlog bezeichnet.

Es ist üblich, dass das Sprint Backlog auf einem Scrumboard oder einer Aufgabentafel dargestellt wird, welche eine dauerhaft sichtbare Abbildung des Status der User Storys im Backlog liefert. Im Sprint Backlog

sind ebenfalls alle Risiken, welche mit den verschiedenen Aufgaben in Verbindung gebracht werden, markiert sowie Aktivitäten zur Risikominderung.

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 **Sprint Burndown Chart***

Beim Sprint Burndown Chart handelt es sich um eine Grafik, die den Umfang der im laufenden Sprint noch zu verrichtenden Arbeit darstellt. Das anfängliche Burndown Chart sollte am Ende eines jeden Tages aktualisiert werden, sobald die Arbeit erledigt ist. Dieses Chart zeigt den vom Scrum Team erreichten Fortschritt und ermöglicht es außerdem, inkorrekte Schätzungen zu entdecken. Wenn das Sprint Burndown Chart anzeigt, dass das Scrum Team die Aufgaben des Sprints in der vorgegebenen Zeit voraussichtlich nicht erledigen kann, dann sollte der Scrum Master sofort jegliche Hindernisse oder Behinderungen identifizieren und versuchen, sie zu beseitigen, um die Arbeit erfolgreich fertigzustellen.

Ein ähnliches Chart ist ein Sprint Burnup Chart. Im Gegensatz zu den Sprint Burndown Charts, welche die Menge der verbleibenden Arbeit anzeigen, stellt das Spint Burnup Chart die abgeschlossene Arbeit als Teil des Sprints dar.

9.7 Phasen Datenflussdiagramm

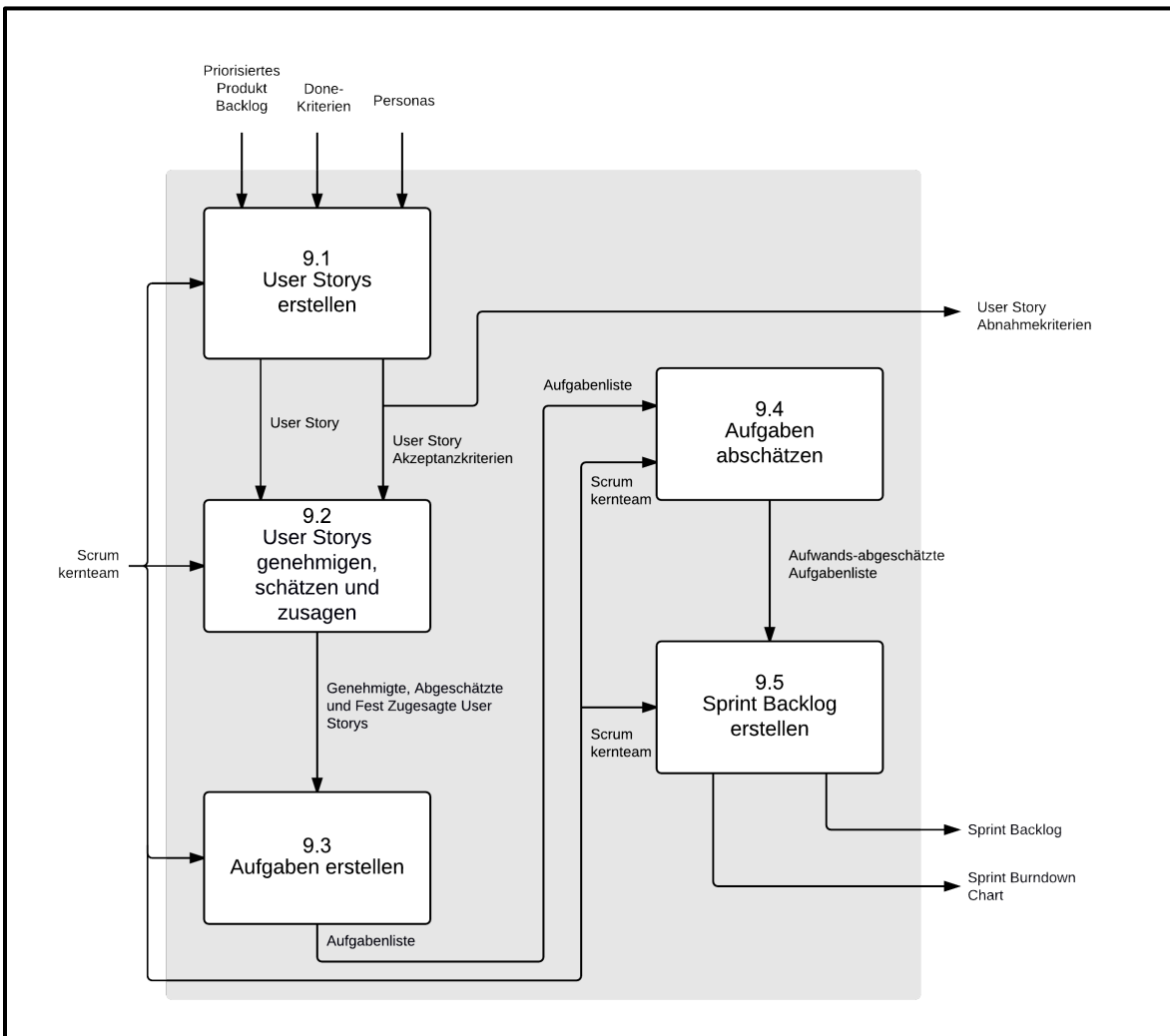


Abbildung 9-14 Planungs- und Schätzungsphase-Datenflussdiagramm

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 Stakeholder 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 Abbildung 10-1 zeigt eine Übersicht über die Prozesse der Implementierungsphase, welche sich folgendermaßen aufteilen:

10.1 Zu liefernde Ergebnisse erzeugen—Während dieses Prozesses arbeitet das Scrum Team an den Aufgaben im Sprint Backlog, um die zu liefernden Ergebnisse des Sprints zu produzieren. Häufig wird dabei

ein Scrumboard verwendet, um die umgesetzten Arbeiten und Aktivitäten zu verfolgen. Angelegenheiten oder Probleme, welche durch das Scrum Team aufgezeigt werden, können in einem Hinweis Log aktualisiert werden.

10.2 Tägliches Standup durchführen—Tägliches Standup Durchführen ist ein Prozess, in dem jeden Tag ein hoch konzentriertes 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.

10.3 Priorisiertes Produkt Backlog pflegen—Das ist ein Prozess, in dem das Priorisierte Produkt Backlog ständig aktualisiert und gepflegt wird. 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.

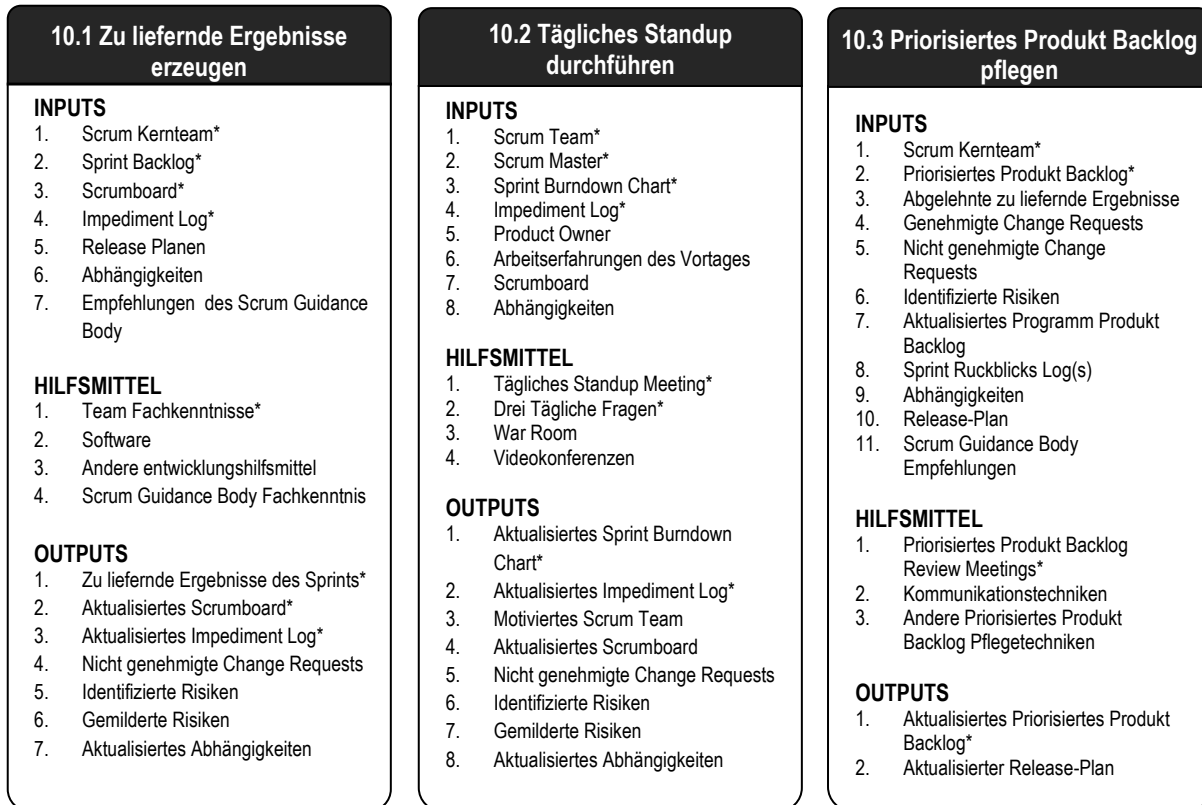


Abbildung 10-1: Implementierungsübersicht

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

Abbildung 10-2 unten zeigt die verbindlichen Inputs, Hilfsmittel und Outputs für Prozesse in der Implementierungsphase.

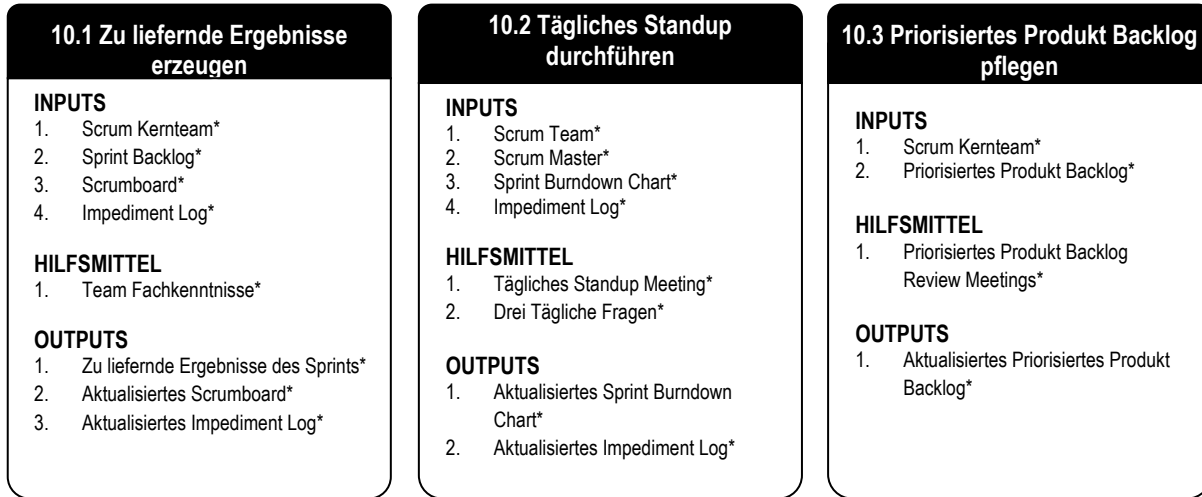


Abbildung 10-2: Implementierungsübersicht (Kernpunkte)

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

10.1 Zu liefernde Ergebnisse erzeugen

Abbildung 10-3 zeigt Inputs, Hilfsmittel und Outputs für den *Zu liefernde Ergebnisse erzeugen* Prozess.

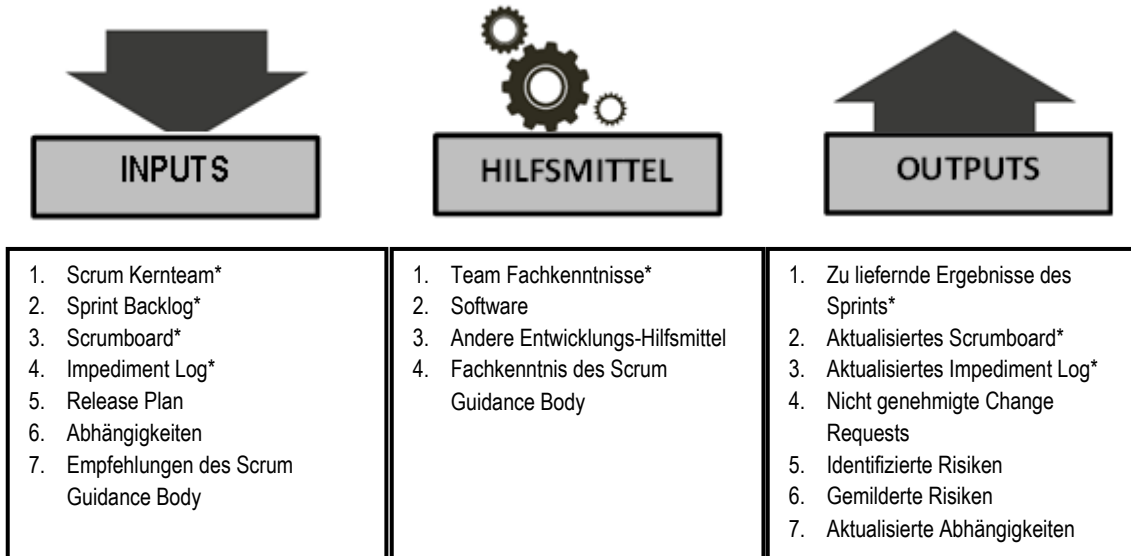


Abbildung 10-3: Zu liefernde Ergebnisse erzeugen–Inputs, Hilfsmittel und Outputs

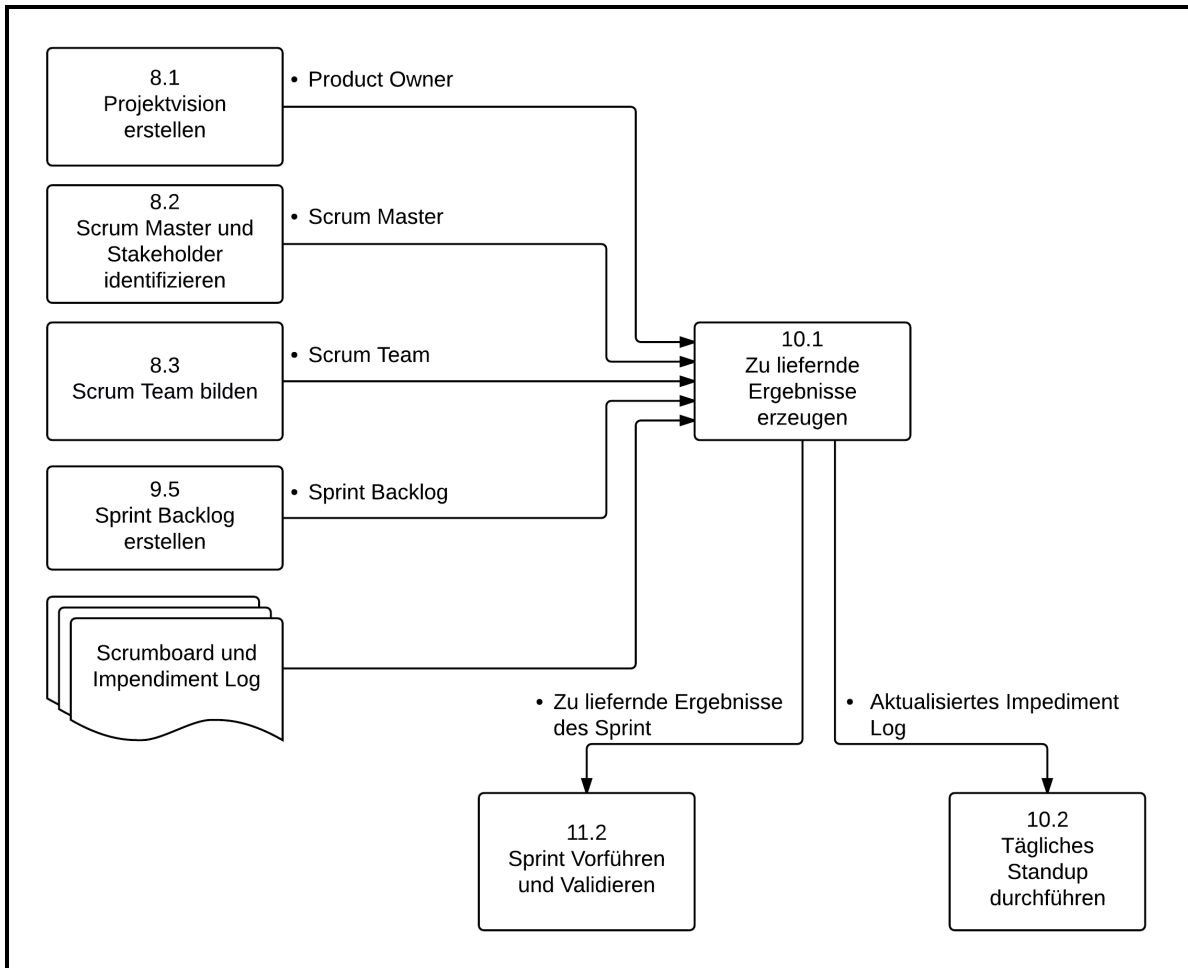


Abbildung 10-4: Zu liefernde Ergebnisse erzeugen–Datenflussdiagramm

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*

Die Transparenz von Scrum kommt von offen sichtbaren Hilfsmitteln zur Informationsverbreitung wie dem Scrumboard, welches den Teamfortschritt anzeigt. Das Team benutzt das Scrumboard zur Planung und Verfolgung der Fortschritte in jedem Sprint. Das Scrumboard enthält vier Spalten, mit denen der Fortschritt der abgeschätzten Aufgaben für den Sprint angegeben wird: eine Spalte „unerledigt“ (To Do) für Aufgaben, die noch nicht begonnen wurden, eine Spalte „In Arbeit“ (in Progress) für Aufgaben die begonnen wurden, aber noch nicht abgeschlossen sind, eine Spalte „Im Test“ (Testing) für Aufgaben die erledigt wurden, sich aber gegenwärtig im Test befinden und eine Spalte „erledigt“ (Done) für die Aufgaben, die erledigt wurden und erfolgreich getestet sind.

Storys	Unerledigt	In Arbeit	Im Test	Erledigt
1			■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
2		■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
3	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
4	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■

Abbildung 10-5: Scrumboard

Vorzugsweise sollte das Scrumboard manuell auf Papier oder auf einem White Board festgehalten werden. Es kann aber auch elektronisch in Form einer Tabelle festgehalten werden.

Das Scrum Team sollte, wenn nötig das Scrumboard anpassen oder Dinge hinzufügen, damit das Scrumboard visuelle Informationen und Kontrolle über die vereinbarten und vom Team zugesagten Arbeitsfortschritte bietet.

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 Planen**

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 Stakeholder außerhalb des Teams involviert. Zusätzliche Expertise ist im Scrum Guidance Body verfügbar, falls sie benötigt wird.

10.1.2.2 Software

Automatisierte Software Hilfsmittel können zur Planung sowie Informationens- Sammlung und Verteilung genutzt werden. Virtuelle Kollaborations Hilfsmittel sind in Projekten unerlässlich, in denen das Scrum Team nicht gemeinsam untergebracht ist. Eine Vielzahl an software-basierten Hilfsmitteln steht zur Verfügung. Sie ermöglichen die Fortschrittsverfolgung, die Sammlung und die Verteilung von Daten und tragen dazu bei, die Prozesse zu beschleunigen.

10.1.2.3 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.

10.1.2.4 Fachkenntnis des Scrum Guidance Body

Siehe Abschnitt 8.4.2.7.

In den Prozessen *Zu liefernde Ergebnisse erzeugen* und *User Storys abschätzen und zusagen* 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.

10.1.3 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 Stories 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

Abbildung 10-6 zeigt Inputs, Hilfsmittel und Outputs für den *Tägliches Standup durchführen* Prozess.

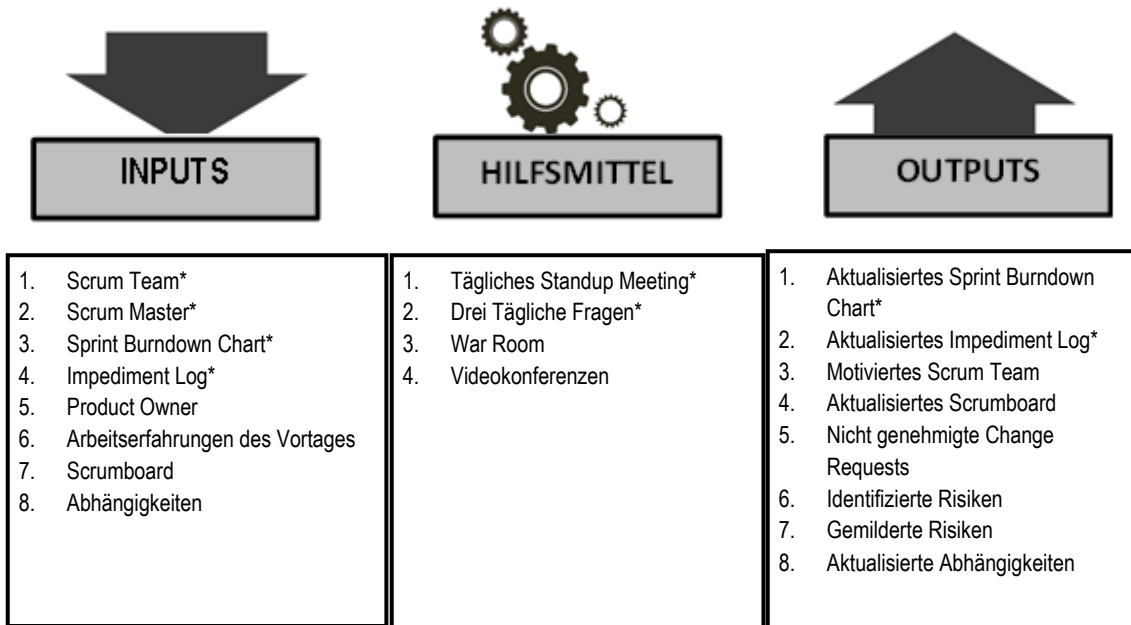


Abbildung 10-6: Tägliches Standup durchführen–Inputs, Hilfsmittel, and Outputs

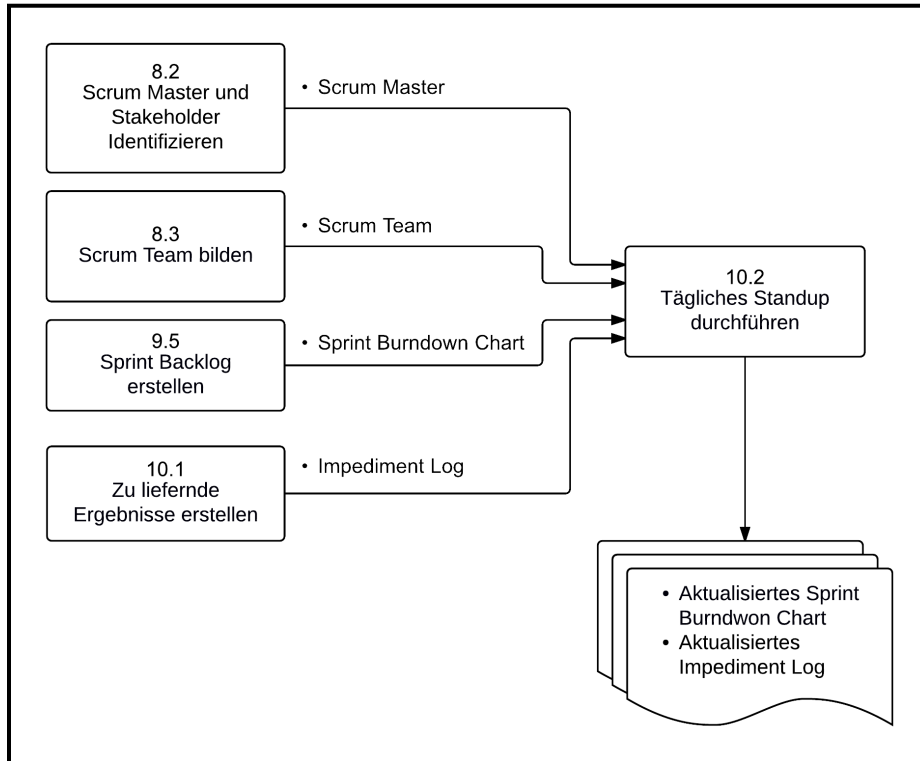


Abbildung 10-7: Tägliches Standup durchführen–Datenflussdiagramm

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

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 Sprint Burndown Chart*

Siehe Abschnitt 9.5.3.2.

10.2.1.4 Impediment Log*

Siehe Abschnitt 10.1.1.4.

10.2.1.5 Product Owner

Siehe Abschnitt 8.1.3.1.

10.2.1.6 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.

10.2.1.7 Scrumboard

Siehe Abschnitt 10.1.1.3.

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.

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.

Stakeholder 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.

10.2.3 Outputs

10.2.3.1 Aktualisiertes Sprint Burndown Chart*

Siehe Abschnitt 9.5.3.2.

10.2.3.2 Aktualisiertes Impediment Log*

Siehe Abschnitt 10.1.1.4.

10.2.3.3 **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.

10.2.3.4 **Aktualisiertes Scrumboard**

Siehe Abschnitt 10.1.1.3.

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 **Aktualisiertes Abhängigkeiten**

Siehe Abschnitt 9.3.3.3.

10.3 Priorisiertes Produkt Backlog pflegen

Abbildung 10-8 zeigt Inputs, Hilfsmittel und Outputs für den *Täglichen Standup durchführen* Prozess.

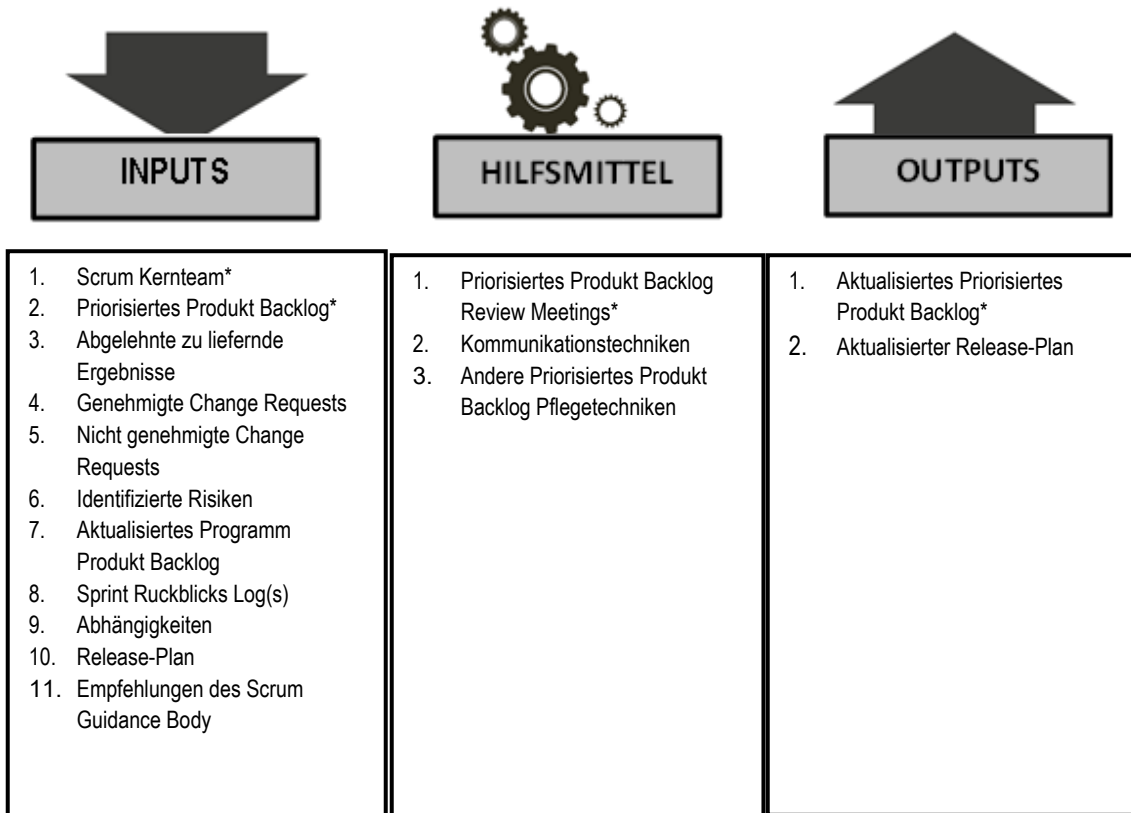


Abbildung 10-8: Priorisiertes Produkt Backlog pflegen–Inputs, Hilfsmittel und Outputs

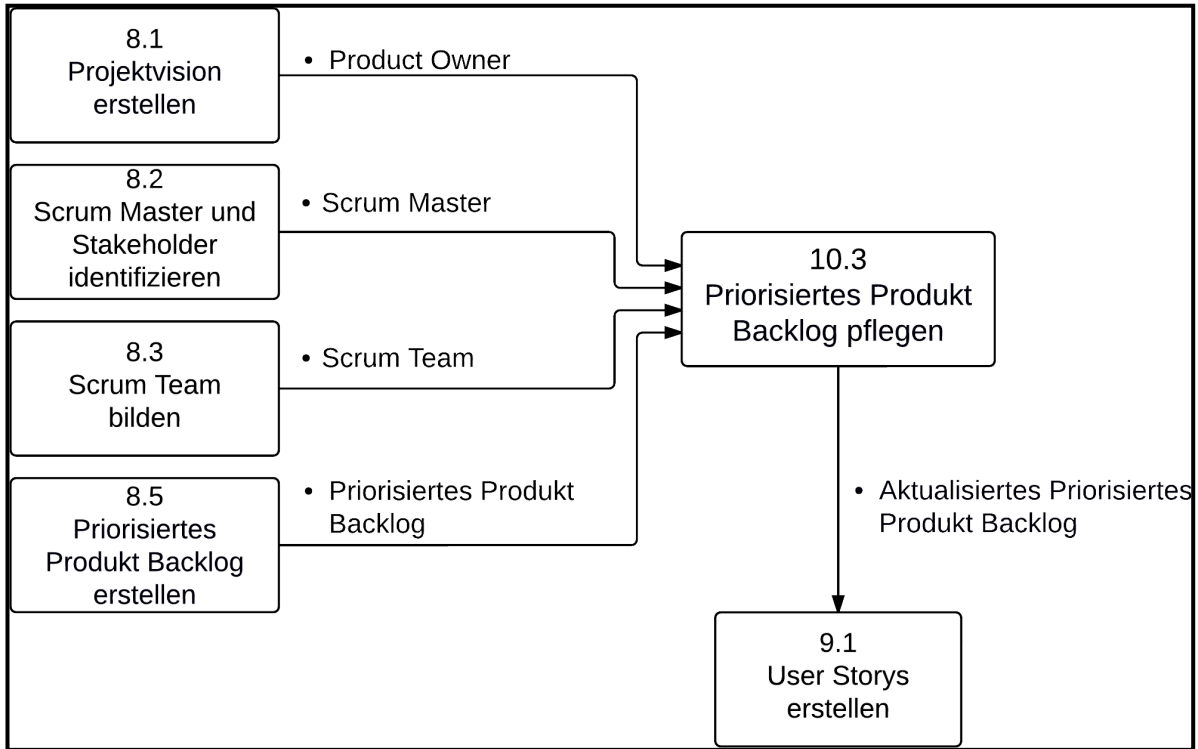


Abbildung 10-9: Priorisiertes Produkt Backlog pflegen–Datenflussdiagramm

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 Abgelehnte zu liefernde Ergebnisse

Wenn ein zu lieferndes Ergebnis nicht den Abnahmekriterien entspricht, wird dieses als abgelehntes zu lieferndes Ergebnis bezeichnet. Abgelehnte zu liefernde Ergebnisse 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 pflegen* wieder priorisiert und für die weitere Entwicklung im nächsten Sprint beachtet werden.

10.3.1.4 Genehmigte Change Requests

Siehe Abschnitt 8.4.1.5.

10.3.1.5 Nicht genehmigte Change Requests

Siehe Abschnitt 8.4.1.6.

10.3.1.6 Identifizierte Risiken

Siehe Abschnitt 8.4.3.4.

10.3.1.7 Aktualisiertes Programm Produkt Backlog

Ähnlich wie beim Projekt Produkt Backlog, wird das Programm Produkt Backlog ebenfalls regelmäßig gepflegt, um Veränderungen und neue Anforderungen zu integrieren. Veränderungen des Programm Produkt Backlogs können aus Änderungen entweder externer oder interner Umstände hervorgehen. Externe Umstände können sich verändernde Geschäftsszenarien, technologische Trends oder rechtliche Konformitätsanforderungen sein. Innere Faktoren, die das Programm Produkt Backlog betreffen, können sich auf Änderungen in organisatorischen Strategien oder Richtlinien, identifizierte Risiken und anderen Faktoren beziehen. Änderungen in den Anforderungen im Programm Produkt Backlog wirken sich häufig auf das Projekt Produkt Backlog des zugrundeliegenden Projektes aus. Sie sollten deswegen während des Prozesses *Priorisiertes Produkt Backlog pflegen* in Betracht gezogen werden.

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

Siehe Abschnitt 8.1.1.11.

Während des Prozesses *Priorisiertes Produkt Backlog pflegen* können die Empfehlungen des Scrum Guidance Body die besten Praktiken darüber enthalten, wie man systematisch Anforderungen vom Stakeholder 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.

10.3.2 Hilfsmittel

10.3.2.1 Priorisiertes Produkt Backlog Review Meetings*

Der Product Owner kann mehrere und separate Meetings mit relevanten 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 pflegen* 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 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 Pflorgetechniken

Viele der Andere Priorisiertes Produkt Backlog Pflege Hilfsmittel beinhalten 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

10.3.3 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 Phasen Datenflussdiagramm

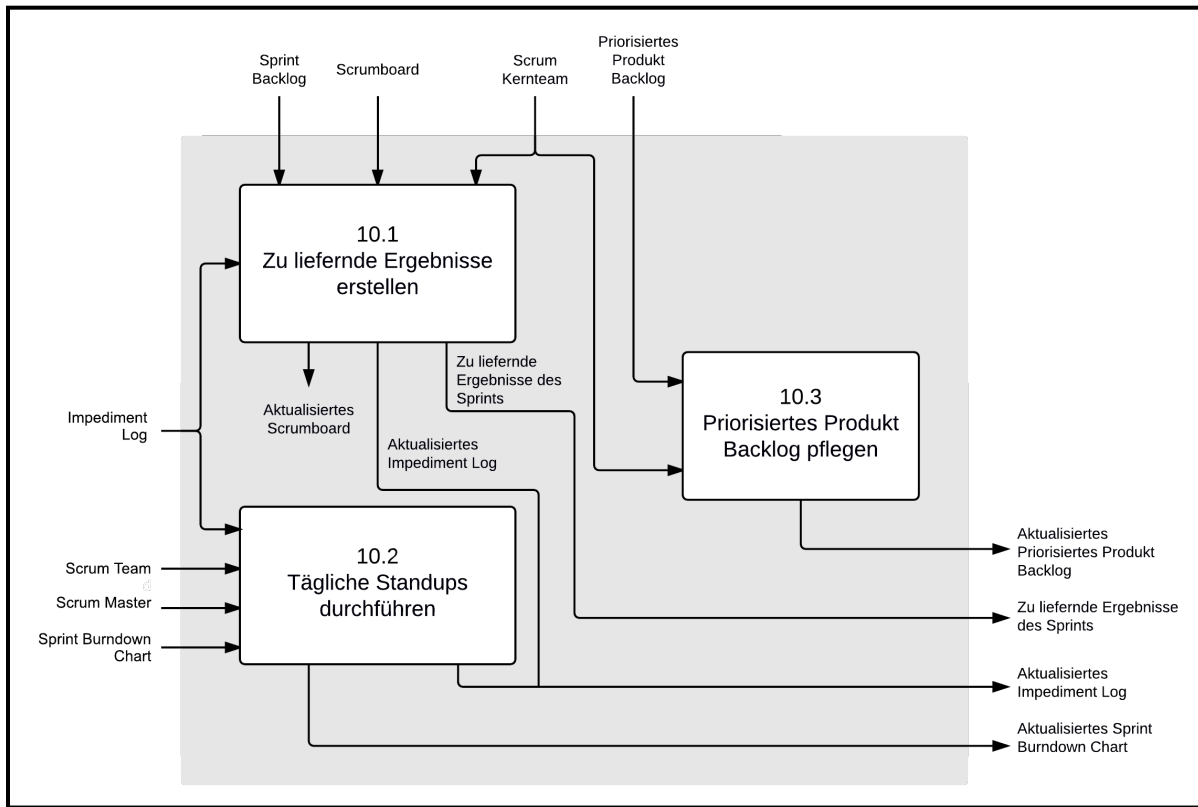


Abbildung 10-10: Implementierungsphase-Datenflussdiagramm

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 Stakeholder 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.

Abbildung 11-1 gibt einen Überblick über die folgenden Vorgänge der Review- und Rückblicksphase:

11.1 Sprint vorführen und validieren-In diesem Prozess zeigt das Scrum Team dem Product Owner und relevanten Stakeholdern die zu liefernden Ergebnisse des Sprints im Rahmen eines Sprint Review Meetings. Das Ziel dieses Treffens ist es, die Zustimmung und Annahme des Produkts oder der Dienstleistung durch den Product Owner zu erlangen.

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.

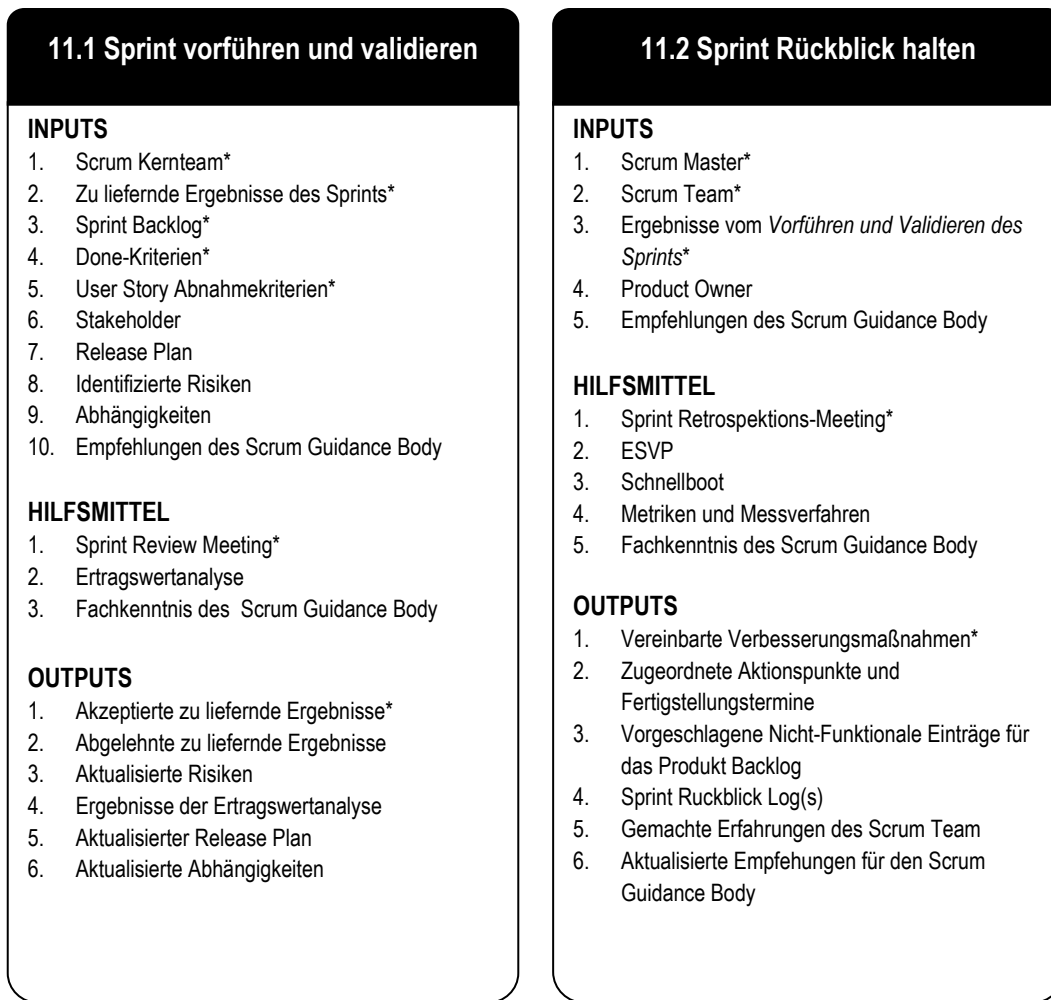


Abbildung 11-1: Überblick der Review- und Rückblicksphase

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess..

Abbildung 11-2 unten zeigt die verbindlichen Inputs, Hilfsmittel und Outputs für Prozesse in Review und Rückblicksphase.

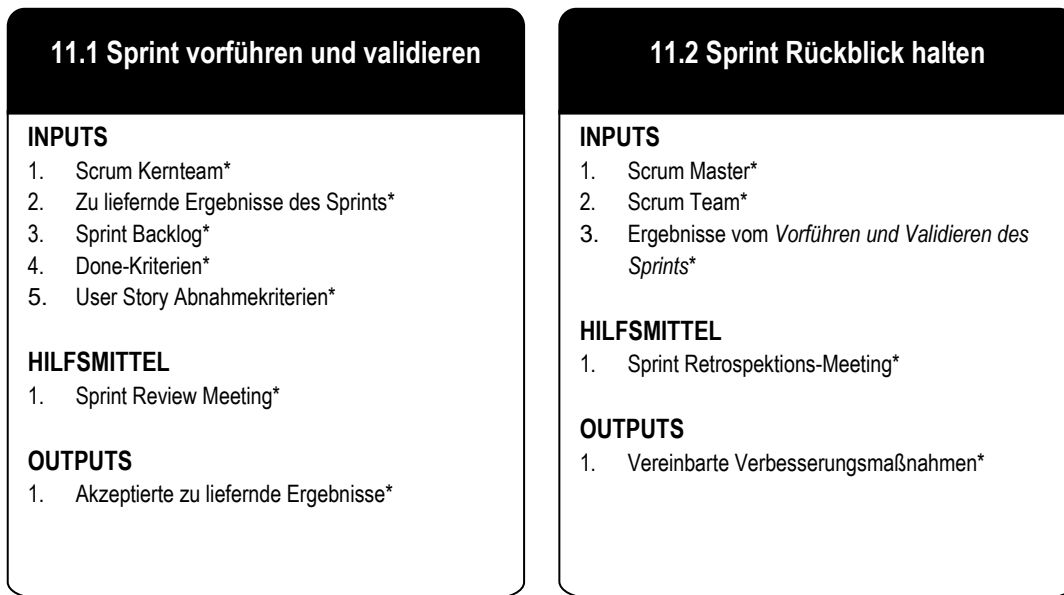


Abbildung 11-2: Überblick der Review und Rückblicksphase (Kernpunkte)

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess..

11.1 Sprint Vorführen und Validieren

Abbildung 11-3 zeigt Inputs, Hilfsmittel und Outputs für den *Sprint Vorführen und Validieren* Prozess.

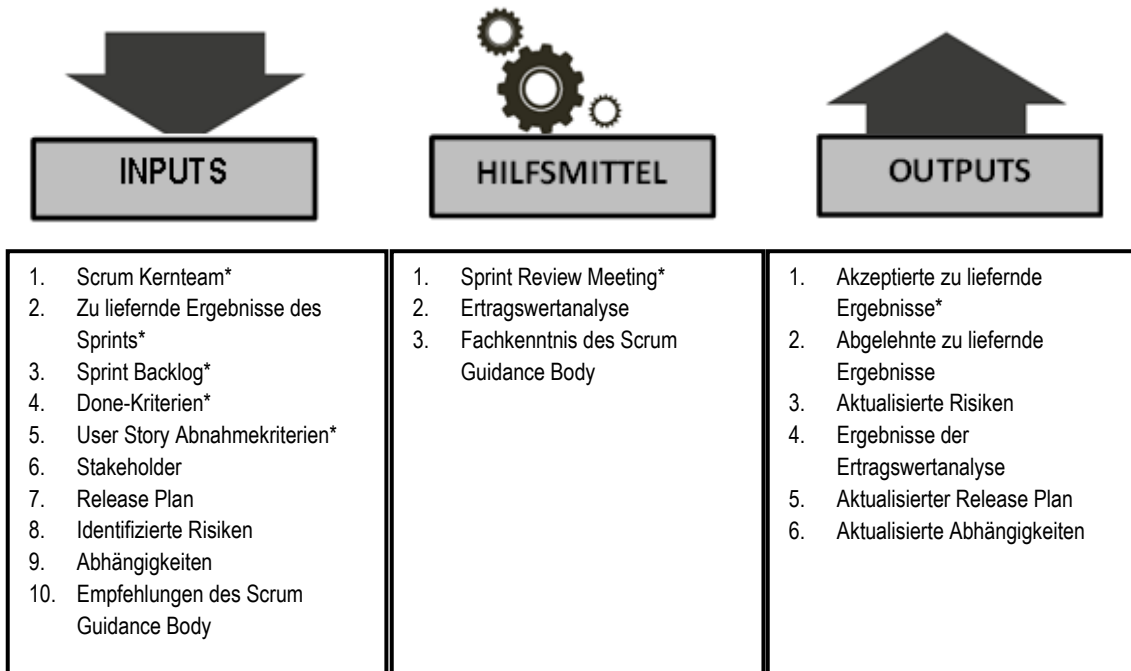


Abbildung 11-3: Sprint Vorführen und Validieren–Inputs, Hilfsmittel und Outputs

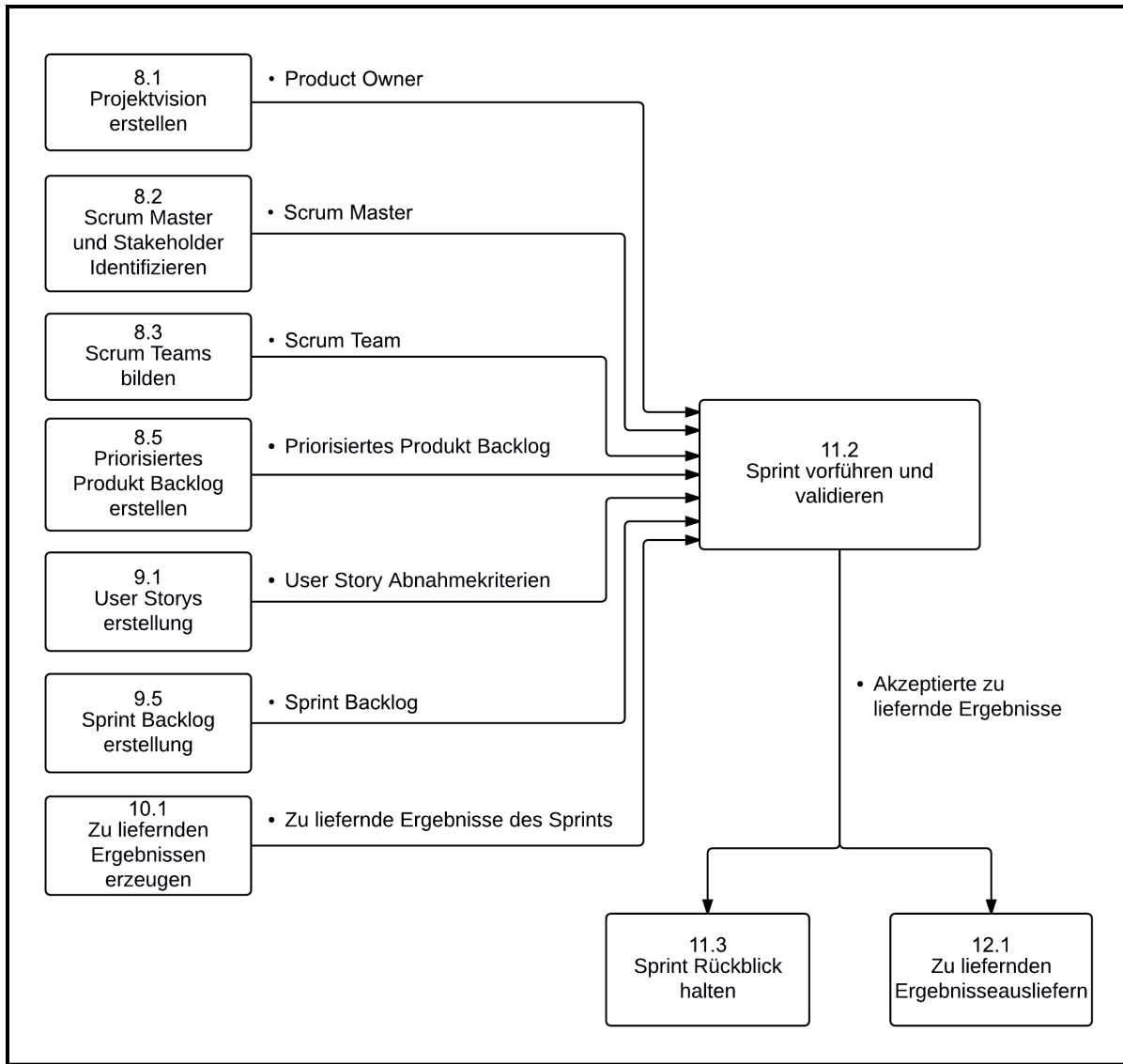


Abbildung 11-4: Sprint Vorführen und Validieren–Datenflussdiagramm

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 Stakeholder

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 Sprint Review Meeting*

Die Mitglieder des Scrum Kernteams und die relevanten Stakeholder nehmen am Sprint Review Meeting teil, um die zu liefernden Ergebnisse, die den User Story Abnahmekriterien entsprechen zu akzeptieren und inakzeptable zu liefernde Ergebnisse abzulehnen. Diese Treffen werden am Ende eines jeden Sprint einberufen. Das Scrum-Team führt die Leistungen aus dem Sprint vor, einschließlich der neuen Funktionalitäten oder erzeugten Produkte. Dies bietet dem Product Owner und den Stakeholdern die Möglichkeit zu prüfen, was bisher fertig gestellt wurde und zu bestimmen, ob in folgenden Sprints irgendwelche Änderungen hinsichtlich des Projekts oder der Prozesse durchgeführt werden sollen.

11.1.2.2 Ertragswertanalyse

Siehe Abschnitt 4.6.1

11.1.2.3 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.

11.1.3 Outputs

11.1.3.1 Akzeptierte zu liefernde Ergebnisse*

Zu liefernde Ergebnisse, die die User Story Abnahmekriterien erfüllen, werden vom Product Owner akzeptiert. Das Ziel eines Sprint ist es, potenziell versandfähige zu liefernde Ergebnisse oder Teilprodukte zu erstellen, die die Abnahmekriterien erfüllen. Diese gelten als akzeptierte zu liefernde Ergebnisse und 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. Wenn ein zu lieferndes Ergebnis die definierten Abnahmekriterien nicht erfüllt, wird es nicht als akzeptiert gewertet, und wird in der Regel in einem anschließenden Sprint weiter bearbeitet, um alle damit verbundenen Probleme zu beheben. Dies ist höchst unerwünscht, weil es das Ziel eines jeden Sprint ist, die Abnahmekriterien für die zu liefernden Ergebnisse zu erfüllen.

11.1.3.2 Abgelehnte zu liefernde Ergebnisse

Wenn zu liefernde Ergebnisse die Abnahmekriterien nicht erfüllen, werden diese abgelehnt. Die mit den abgelehnten zu liefernden Ergebnissen verbundenen User Storys, werden dem Priorisierten Produkt Backlog wieder hinzugefügt, so dass diese zu liefernden Ergebnisse als Teil eines nachfolgenden Sprints in Betracht gezogen werden können.

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.

11.1.3.6 Aktualisierte Abhängigkeiten

Siehe Abschnitt 9.3.3.3.

11.2 Sprint Rückblick halten

Abbildung 11-5 zeigt Inputs, Hilfsmittel und Outputs für den *Sprint Rückblick halten* Prozess.

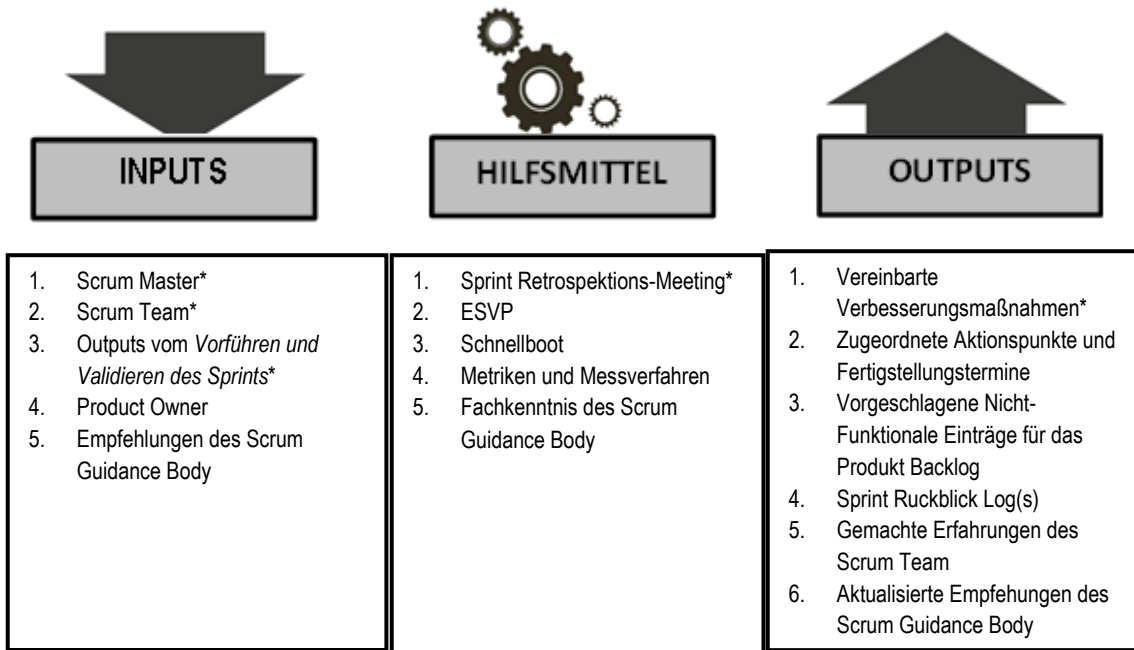


Abbildung 11-5: Sprint Rückblick halten-Inputs, Hilfsmittel und Outputs

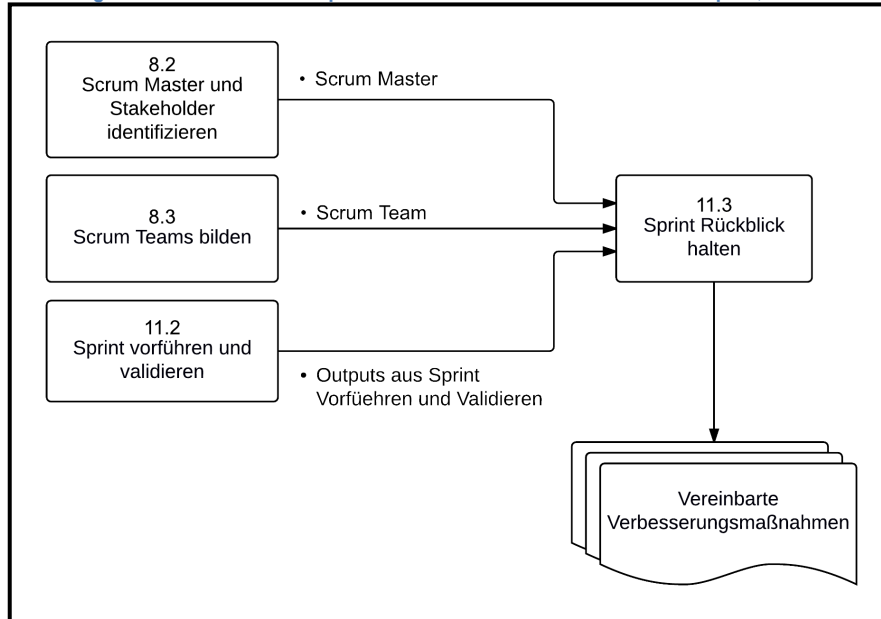


Abbildung 11-6: Sprint Rückblick halten- Datenflussdiagramm

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess..

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 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.

11.2.3 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, dass es aus allen Fehlern, die während eines Sprints 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 Phase

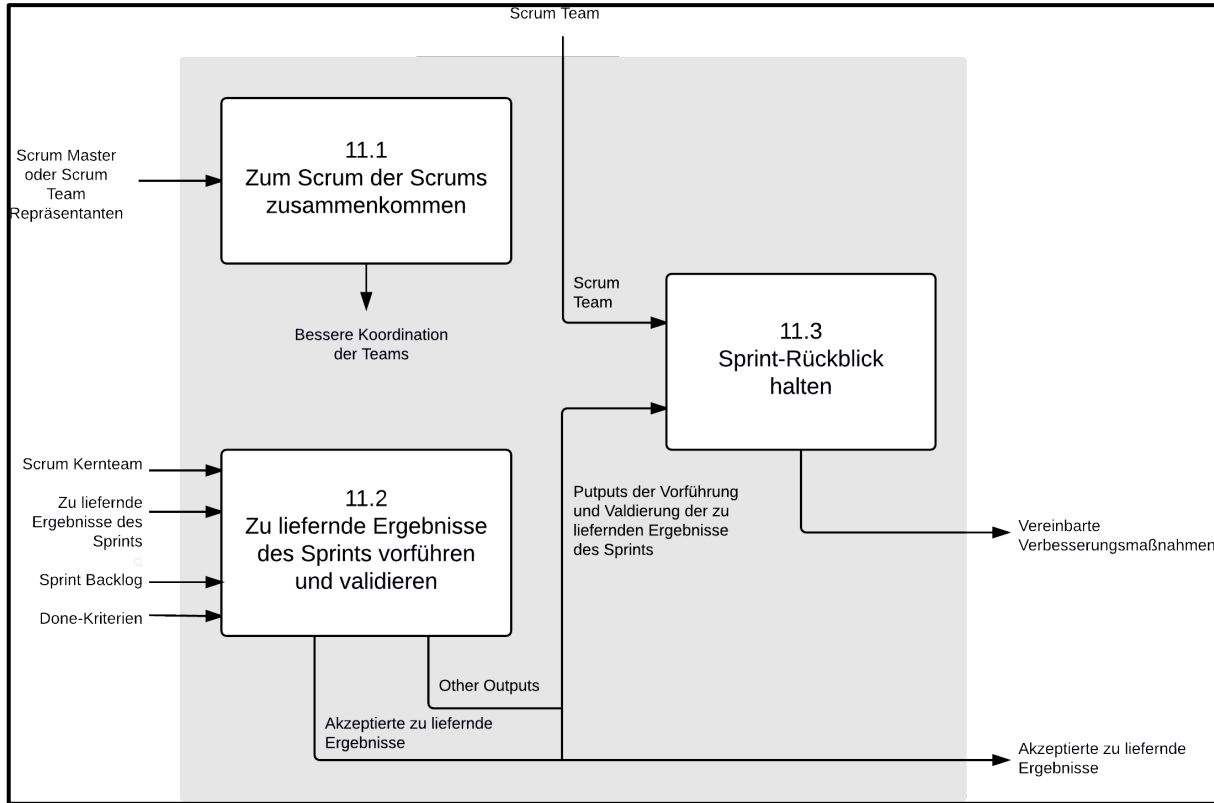


Abbildung 11-3: Übersicht und Rückblicksphase-Datenflussdiagramm

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 Stakeholder 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.

Abbildung 12-1 zeigt eine Übersicht über die Prozesse der Releasephase, welche sich folgendermaßen aufteilen:

12.1 Zu liefernde Ergebnisse ausliefern—In diesem Prozess werden Akzeptierte zu liefernde Ergebnisse ausgeliefert oder an den zuständigen Stakeholder weitergeleitet. Eine formale Vereinbarung über Funktionierende zu liefernde Ergebnisse dokumentiert den erfolgreichen Abschluss des Sprints.

12.2 Projekt Rückblick halten—In diesem Prozess, der das Projekt abschließt, kommen Stakeholder 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.

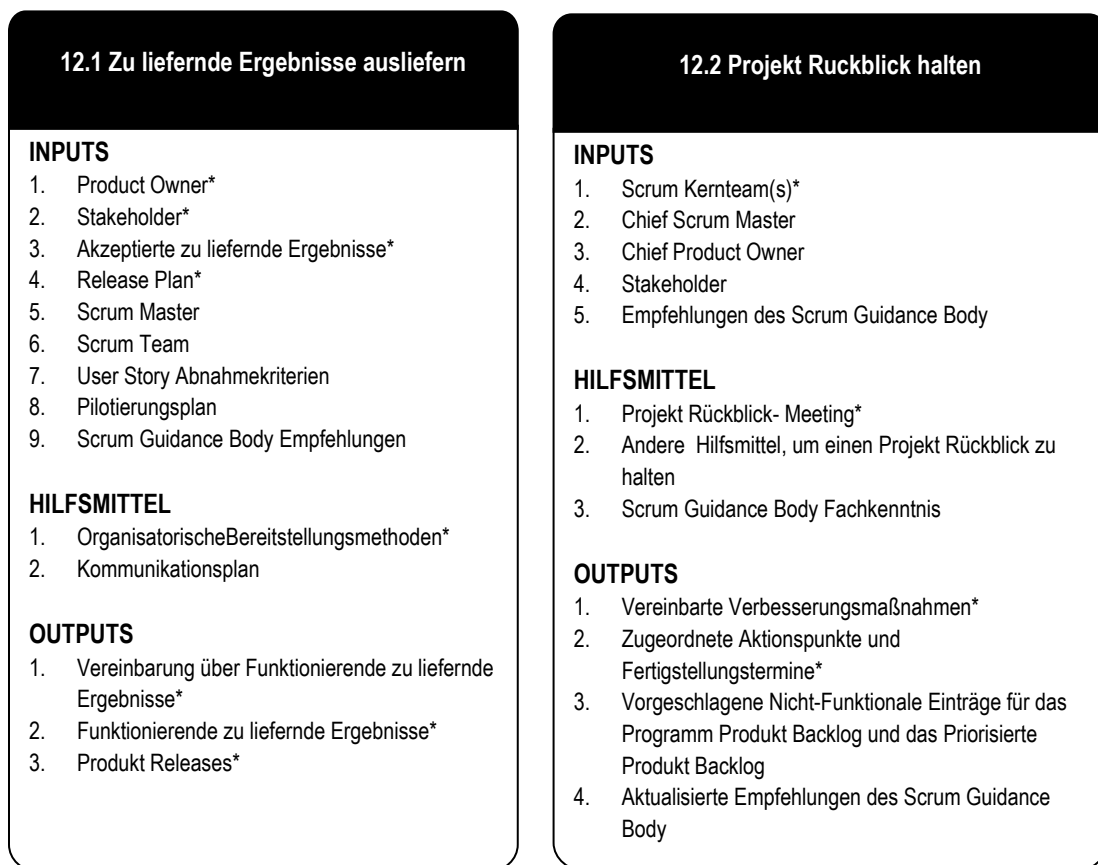


Abbildung 12-1: Release Übersicht

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

Abbildung 12-2 unten zeigt die verbindlichen Inputs, Hilfsmittel und Outputs für Prozesse in Releasephase.

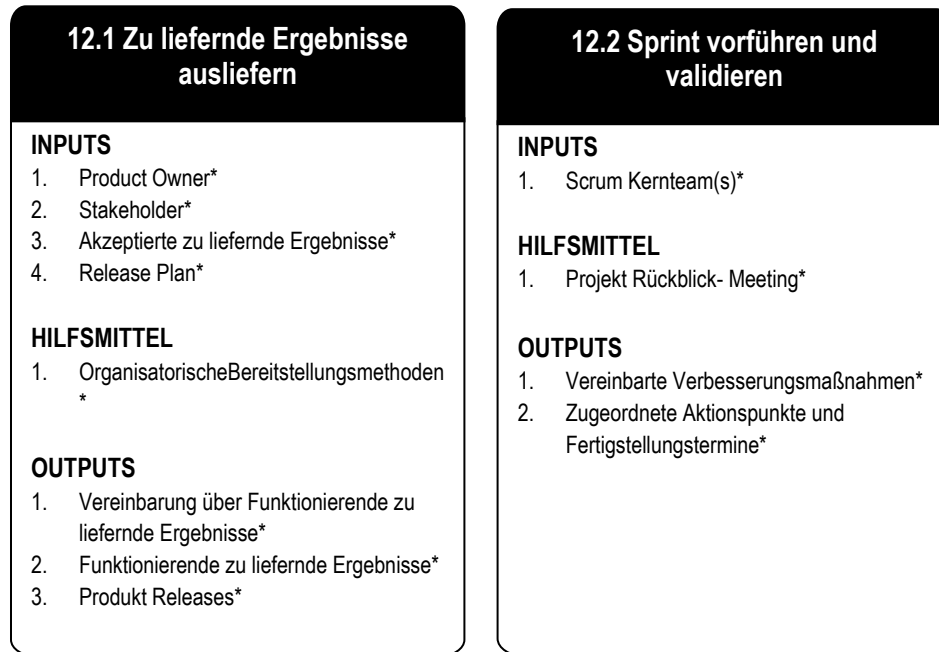


Abbildung 12-2: Release Übersicht (Kernpunkte)

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

12.1 Zu liefernde Ergebnisse ausliefern

Abbildung 12-3 zeigt Inputs, Hilfsmittel und Outputs für den *Zu liefernde Ergebnisse ausliefern* Prozess.

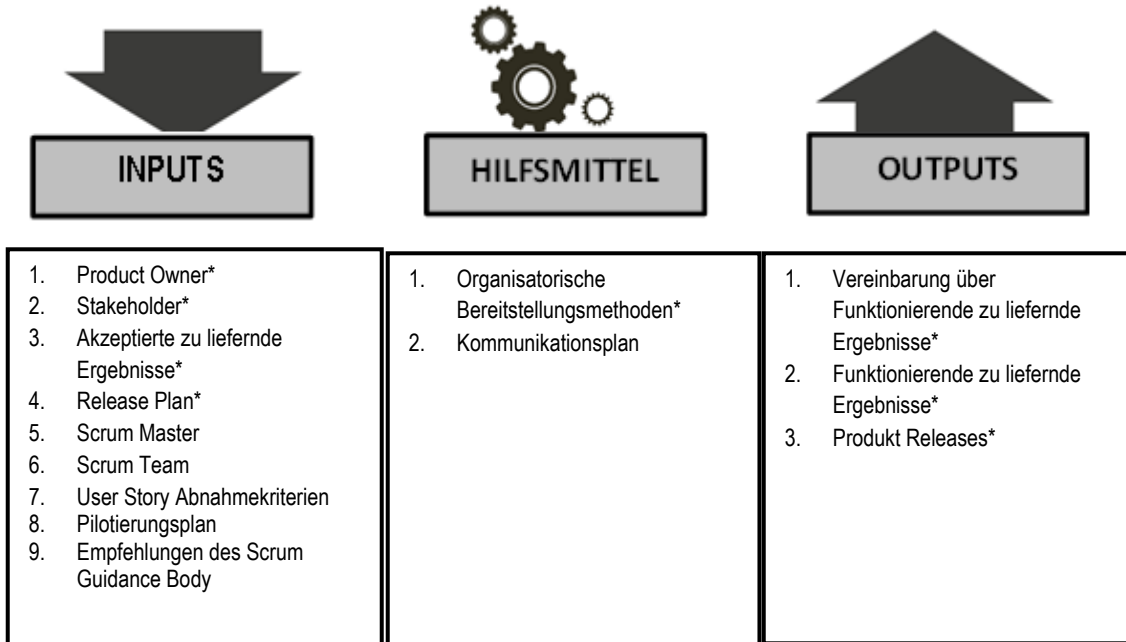


Abbildung 12-3: Zu liefernde Ergebnisse ausliefern–Inputs Hilfsmittel und Outputs

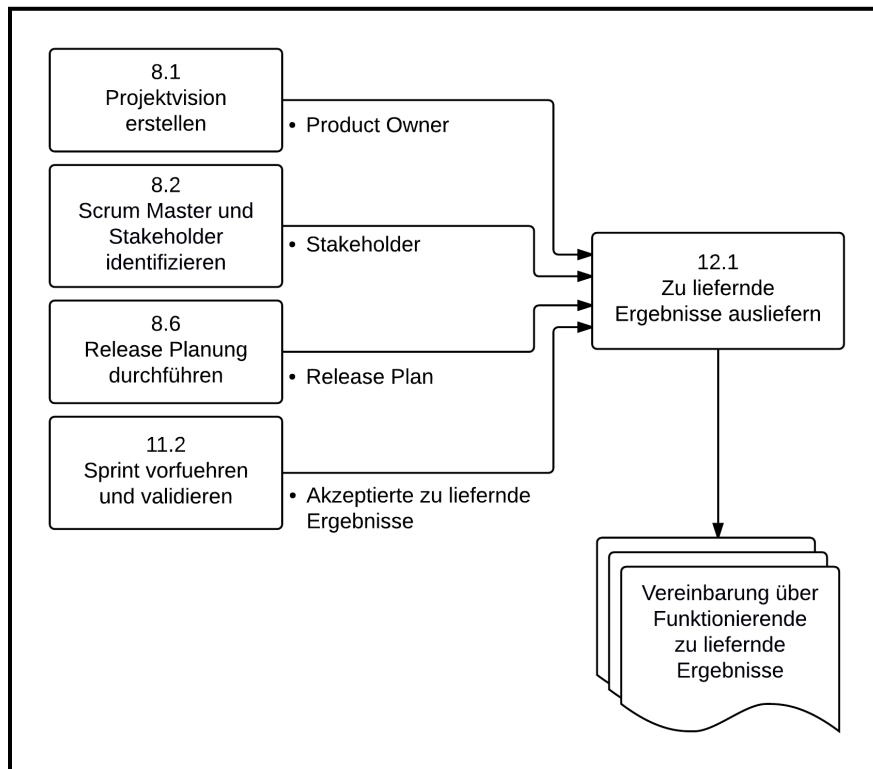


Abbildung 12-4: Zu liefernde Ergebnisse ausliefern – Datenflussdiagramm

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

12.1.1 Inputs

12.1.1.1 Product Owner*

Siehe Abschnitt 8.1.3.1.

12.1.1.2 Stakeholder*

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 Stakeholdern mitgeteilt.

12.1.1.9 **Empfehlungen 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, Zielanwender 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.

12.1.2.2 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 Stakeholder 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.

12.1.3 Outputs

12.1.3.1 Vereinbarung über Funktionierende zu liefernde Ergebnisse*

Zu liefernde Ergebnisse, welche die Abnahmekriterien erfüllen, erhalten geschäftliche Abnahme und Genehmigung durch den Kunden oder den Sponsoren. Die formale Annahme durch den Kunden ist entscheidend für die Wahl der Umsatzrealisierung. Die Verantwortung diese zu erhalten wird in den Unternehmensrichtlinien festgelegt und der Verantwortliche ist nicht unbedingt der Product Owner.

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 Projekt Rückblick halten

Abbildung 12-5 zeigt um so Inputs, Hilfsmittel und Outputs für *Projekt Rückblick halten* Prozess.

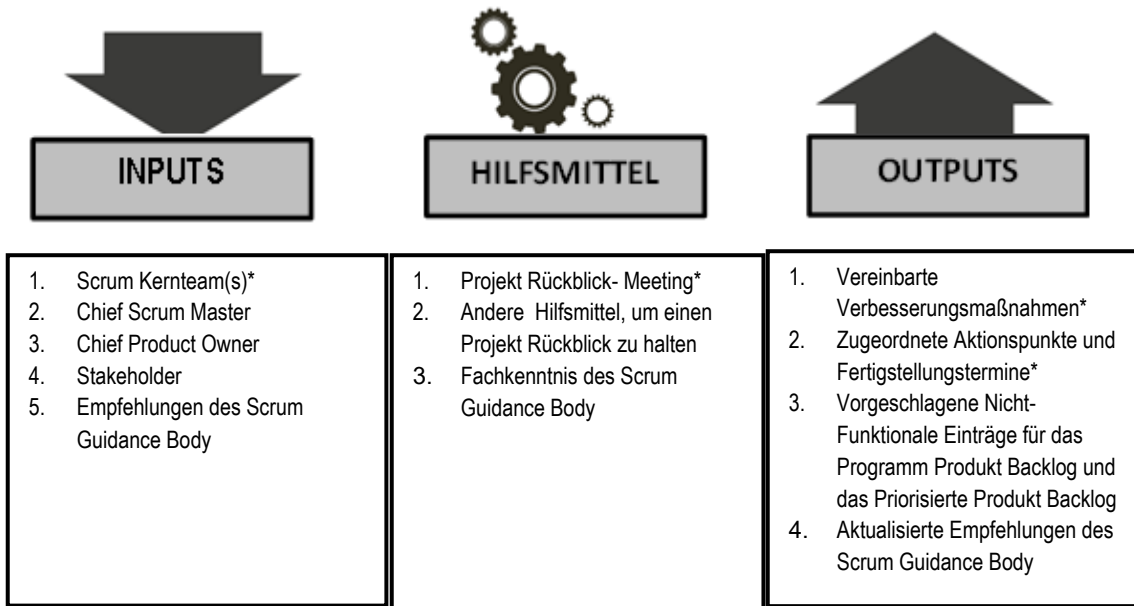


Abbildung 12-5: Projekt Rückblick halten–Inputs, Hilfsmittel und Outputs

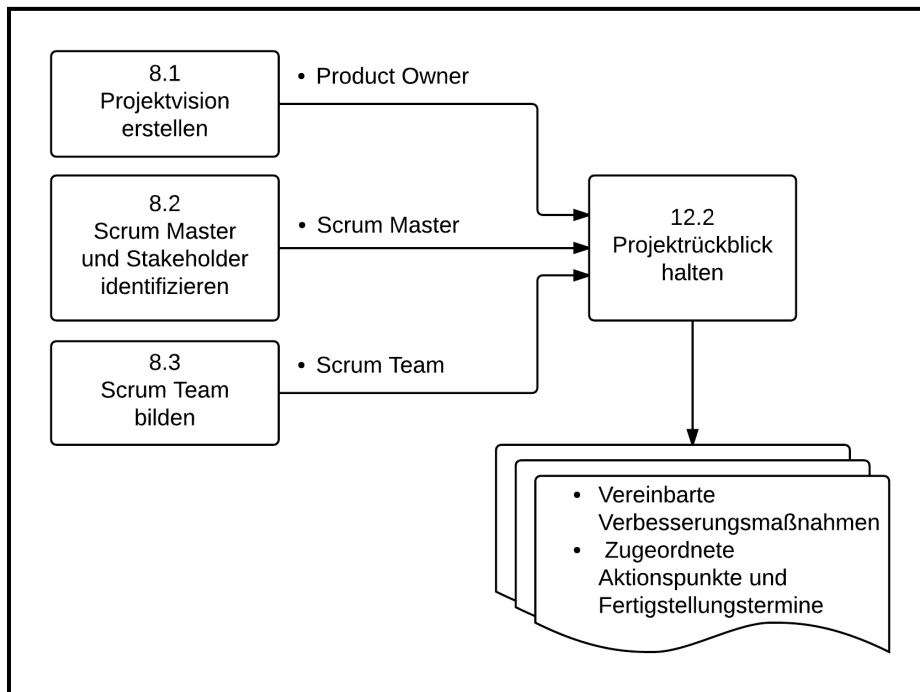


Abbildung 12-6: Projekt Rückblick halten–Datenflussdiagramm

Vermerken: Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich für den entsprechenden Prozess.

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 8.2.1.6.

12.2.1.3 Chief Product Owner

Siehe Abschnitt 8.1.1.5.

12.2.1.4 Stakeholder

Siehe Abschnitt 8.2.3.2.

12.2.1.5 Empfehlungen des Scrum Guidance Body

Siehe Abschnitt 8.1.1.11.

Im Prozess *Projekt 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 Projektrückblick 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 Projekt Rückblick- Meeting*

Das Projekt 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 Stakeholder. 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 Projekt 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 *Projekt 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 *Projekt 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.

12.2.3 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-funktionale 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 Phasen Datenflussdiagramm

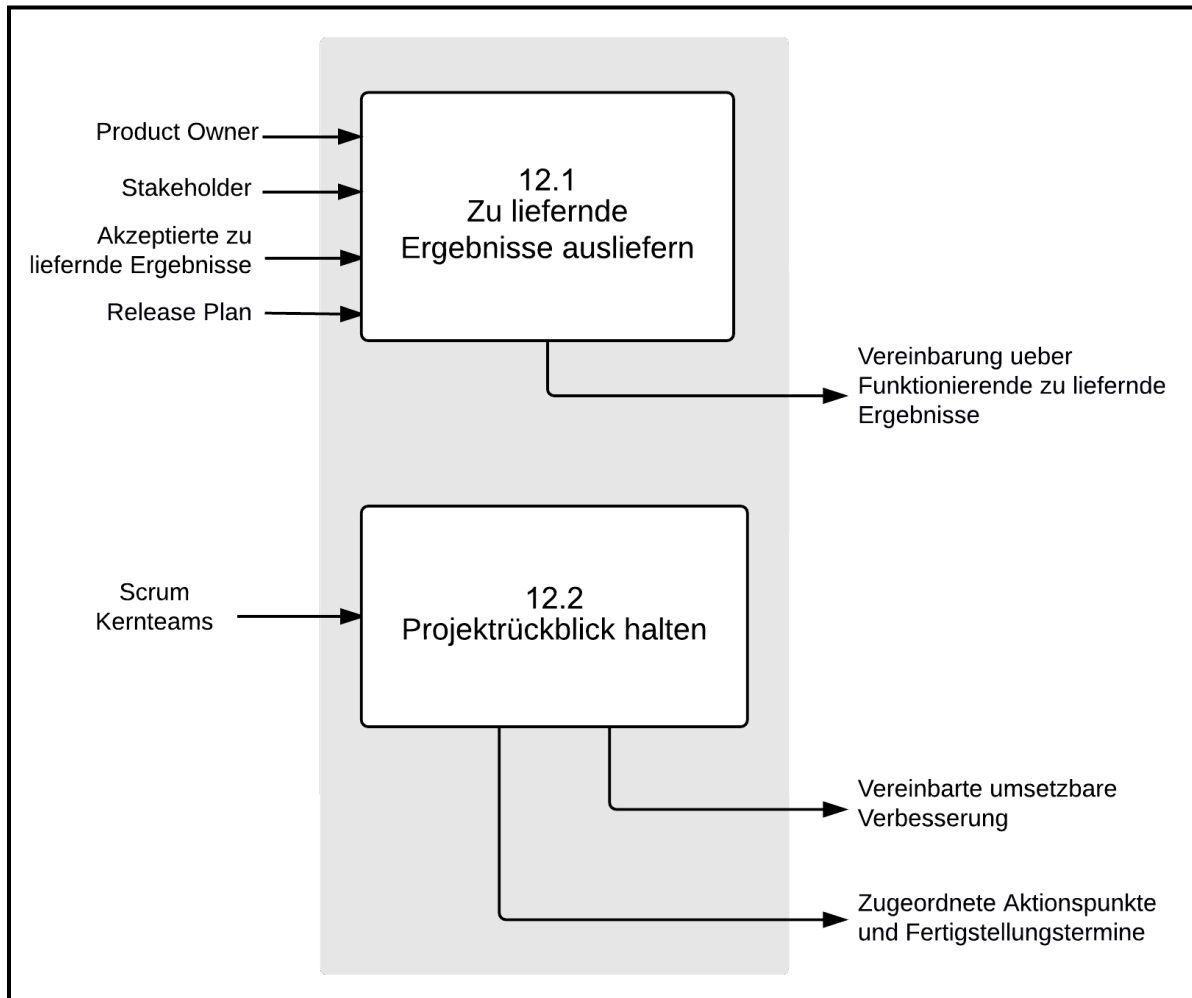


Abbildung 12-7: Releasephase-Datenflussdiagramm

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

Große Projekte in jeder Branche

Produkte, Dienstleistungen oder jegliche an Stakeholder zu liefernde Ergebnisse

Der Begriff "Produkt" in dieser Wissenssammlung 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 mehreren Hundert Team-Mitgliedern und vielen Teams.

Zusätzlich zu den Auswirkungen, die ein großes Projekt auf die fundamentalen Scrum Prozesse in Kapiteln 8 bis 12 hat, werden in diesem Kapitel drei weitere Prozesse eingeführt, die bei großen Projekten zur Anwendung kommen.

Um die bestmögliche Anwendung des Scrum Rahmens zu ermöglichen, präsentiert dieses Kapitel Inputs, Hilfsmittel und Outputs für jeden Prozess als „verbindlich“ oder „optional.“ Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind kritisch für den Erfolg eines Projekts, während solche ohne Sternchen optional sind.

Es wird empfohlen, dass Personen, welche sich erst mit dem Scrum Rahmen und seinen Prozessen vertraut machen, sich primär mit den verbindlichen Inputs, Hilfsmitteln und Outputs beschäftigen, während Chief Product Owner, Product Owner, Chief Scrum Masters, Scrum Masters und Scrum Praktiker mit größerer Erfahrung danach streben sollten, sich ein noch tieferes Wissen über die Informationen in diesem gesamten Kapitel anzueignen. Es ist auch wichtig, festzustellen, dass Prozesse, obwohl sie im *SBOK™* einzeln aufgeführt sind, nicht notwendigerweise sequentiell oder getrennt ausgeführt werden. Manchmal mag es entsprechend den spezifischen Erfordernissen eines jeden Projekts angemessener sein, einige Prozesse zu kombinieren.

Diese Kapitel ist aus dem Blockwinkel eines großen Projektteams geschrieben, das mehrere Scrum Teams in einem großen Projekt koordiniert, mit dem Ziel, potentiell lieferbare Produkt Inkremente / Zu Liefernde Ergebnisse zu erzeugen. Zusätzliche Informationen zur Anwendung von Scrum, nicht nur für große Projekte, ist in den Kapiteln 2 bis 7 enthalten, die Scrum Prinzipien und Aspekte behandeln.

Großes Projekt vs. Typisches Scrum Projekt

Die in Kapiteln 8-12 definierten grundlegenden Scrum Prozesse sind für alle Scrum Projekte gültig, und in jenen Kapiteln erwähnte Konzepte reichen aus, um Scrum Projekte mit nur einigen Scrum Teams - normalerweise 1-3 Scrum Teams - durchzuführen. Auswirkungen auf diese grundlegenden Scrum Prozesse, die nur für große Projekte anwendbar sind, sind am Ende dieses Kapitels beschrieben.

Wenn wir es mit großen Projekten zu tun haben, die im Allgemeinen vier oder mehr Scrum Teams einschließen, können außer den in Kapiteln 8-12 definierten Prozessen noch einige zusätzliche Prozesse notwendig sein, um die zusätzlichen Koordinierungs- und Synchronisationsbemühungen zu adressieren. Die Definition eines großen Projekts kann von der Gesellschaft und der Komplexität der durchgeführten Projekte abhängen. Das entscheidende Kriterium dafür, dass ein Projekt groß ist, ist, dass es mehrere Scrum Master und/oder Product Owner hat.

Einige Gründe, warum wir für große Projekte zusätzliche Prozesse brauchen könnten, sind wie folgt:

- Gesteigerte Interaktion und Abhängigkeiten zwischen Scrum Teams, da die Komplexität bei einem großen Projekt erheblich zunimmt
- Die Notwendigkeit der Zusammenarbeit in einem Team von Product Ownern
- Die Notwendigkeit zur Handhabung von Konflikten, zur Lösung von Problemen und das Setzen von Prioritäten zwischen allen Scrum Teams
- Die Erfordernis zur Spezialisierung, da einige Scrum Teams gegebenenfalls spezialisierte Ressourcen für bestimmte Aufgaben im großen Projekt erfordern - und diese Fähigkeiten müssen nicht in allen Scrum Teams verfügbar sein.
- 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 Industrien). Diese müssen gegebenenfalls im Scrum Guidance Body definiert werden.
- Die Erfordernis, eine Umgebung für das große Projekt einzurichten, die dann von allen Scrum Teams verwendet wird.
- Der Bedarf zur Koordinierung der Outputs von mehreren Scrum Teams, um ein Release für ein großes Projekt zu erzeugen.

Die zusätzlichen Prozesse für die Skalierung für große Projekte sind die Folgenden:

13.1 Komponenten des Großen Projekts Erstellen-dieser Prozess definiert wie die verschiedenen Product Owner als Team zusammenarbeiten, und wie die verschiedenen Scrum Teams zusammenarbeiten. Außerdem werden gemeinsame Komponenten Bestandteile sowie gemeinsame und spezialisierte Ressourcen identifiziert.

13.2 Sprints Durchführen und Koordinieren-dieser Prozess ist nur für große Projekte relevant und spricht bestimmte Aspekte an, die während jedes Sprints in Erwägung gezogen werden sollten. Falls erforderlich, werden Scrum der Scrums Meetings durchgeführt, um die Arbeiten zwischen verschiedenen Scrum Teams zu koordinieren.

13.3 Ein Release im Großen Projekt Vorbereiten-In einigen großen Projekten kann es geschäftlich Sinn machen, vor einem Release einen speziellen Sprint zur Vorbereitung dieses Releases durchzuführen (das wird vom Projekt Team entsprechend den geschäftlichen Notwendigkeiten entschieden). Dieser Prozess behandelt einen solchen Vorbereitungs-Sprint. Die Existenz eines Vorbereitungs-Sprints darf nicht dazu führen, dass Aktivitäten, die in die anderen Sprints gehören, bis zu diesem Punkt verschoben werden.

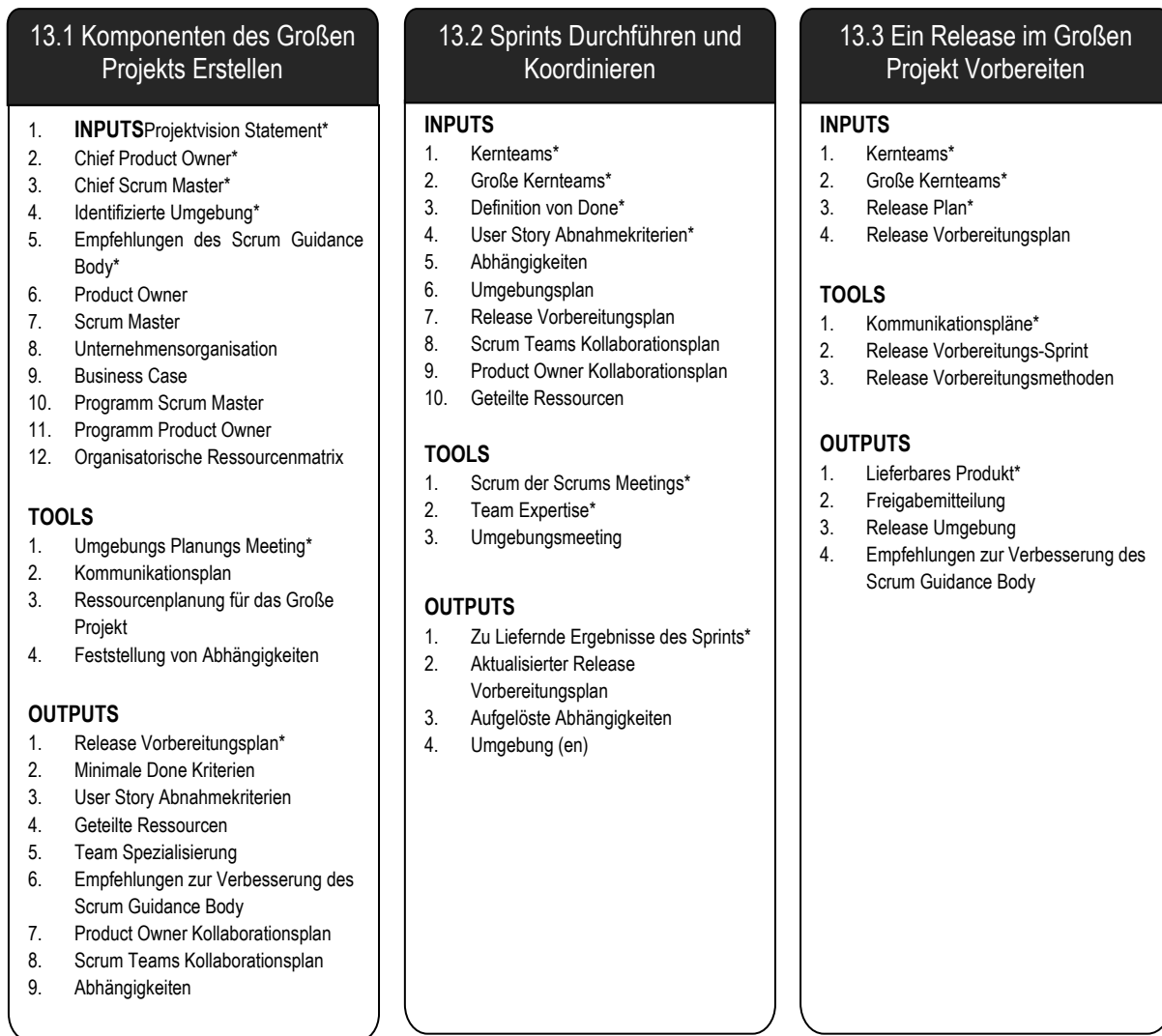


Abbildung 13-1: Scrum für große Projekte Skalieren

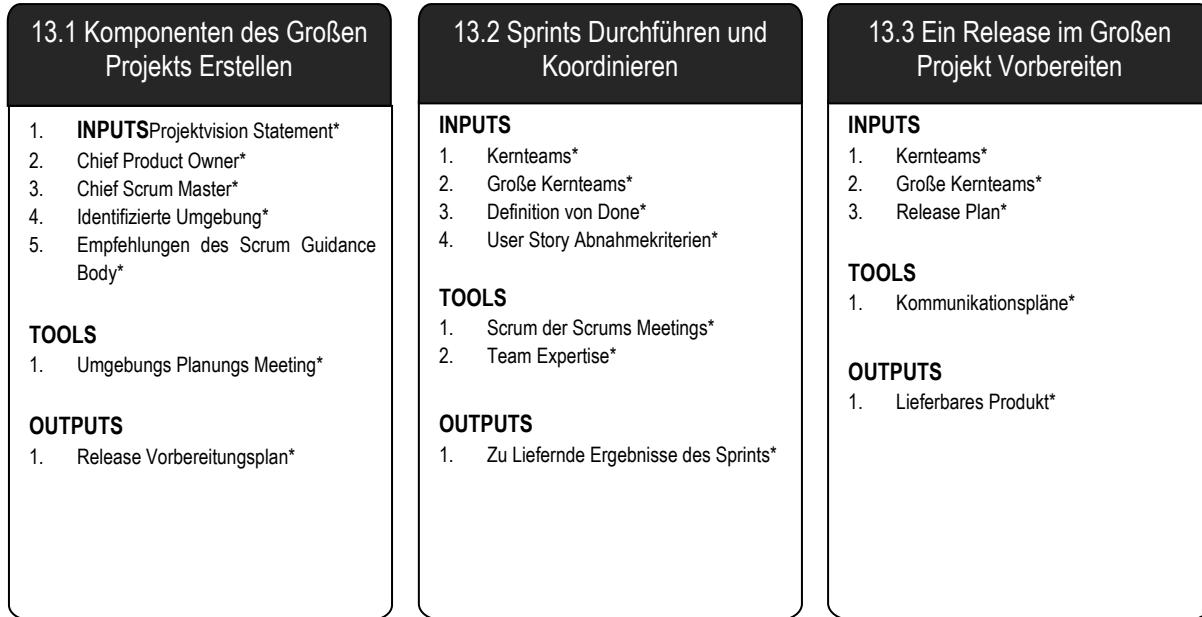


Abbildung 13-2: Scrum für große Projekte Skalieren

13.1 Komponenten des Großen Projekts Erstellen

Abbildung 13-3 zeigt alle Inputs, Hilfsmittel und Outputs für den *Komponenten des Großen Projekts Erstellen* Prozess.

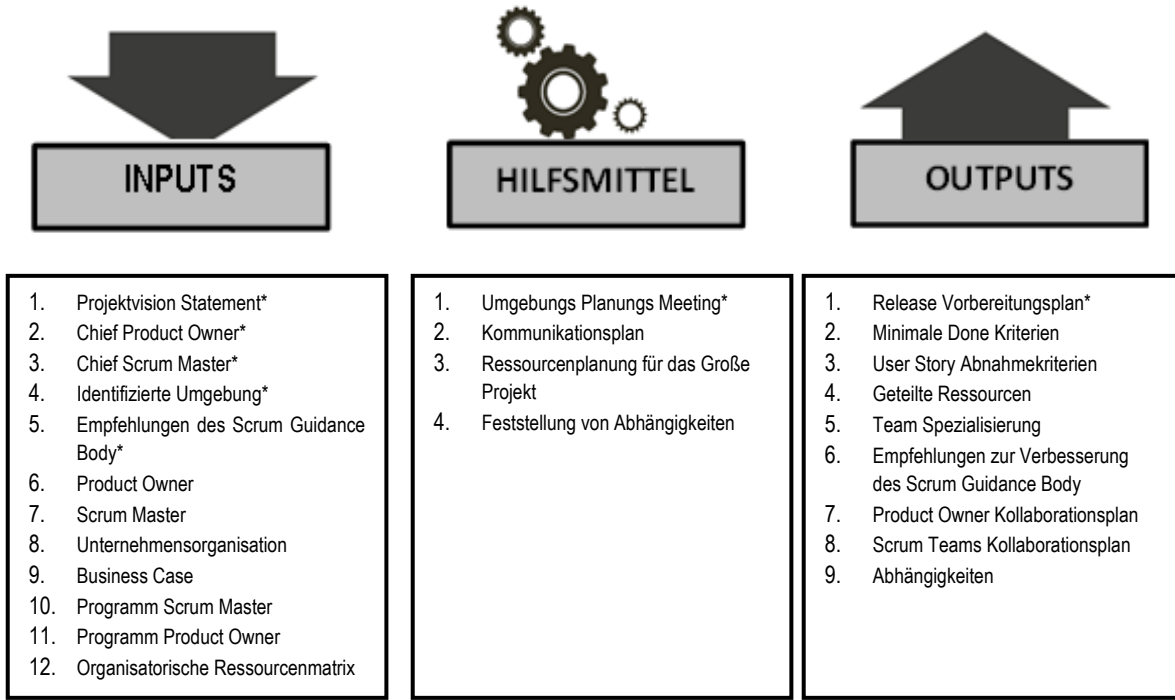


Abbildung 13-3: Programm- oder Portfolio- Komponenten Erstellen - Inputs, Hilfsmittel und Outputs

Hinweis - Sternchen (*) bezeichnen einen "verbindlichen" Input, ein "verbindliches" Hilfsmittel oder einen "verbindlichen" Output für den entsprechenden Prozess.

Abbildung 13-4 zeigt alle Beziehungen des *Programm- oder Portfolio- Komponenten Erstellen* Prozesses zu den grundlegenden Scrum Prozessen der Initiierungsphase.

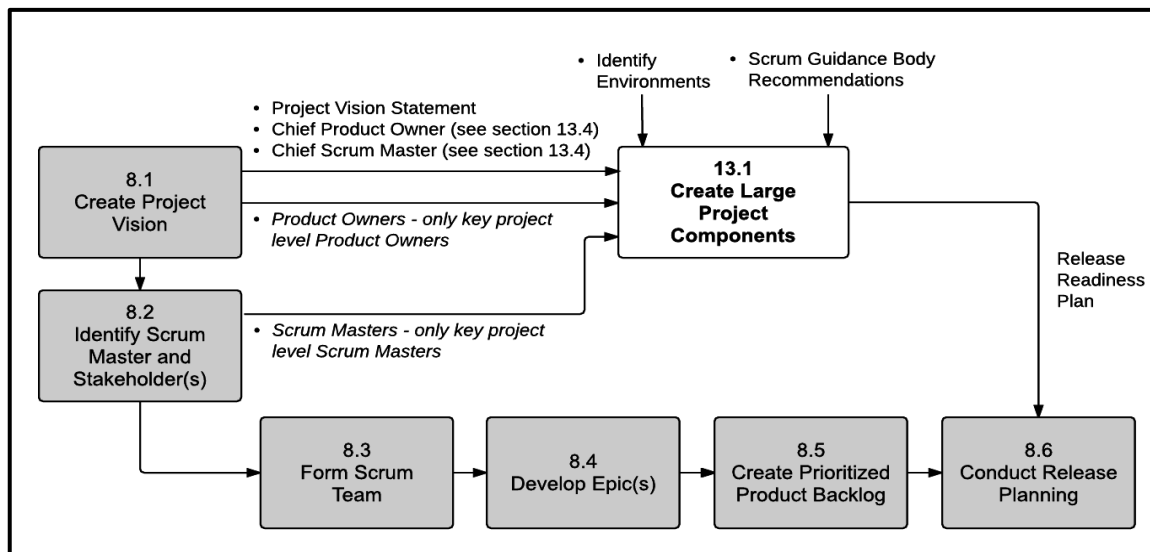


Abbildung 13-4: Programm- oder Portfolio- Komponenten Erstellen - Datenflussdiagramm

13.1.1 Inputs

13.1.1.1 Projekt Visions Statement

Beschrieben in Abschnitt 8.1.3.2.

13.1.1.2 Chief Product Owner*

Beschrieben in Abschnitt 3.4.2.

13.1.1.3 Chief Scrum Master*

Beschrieben in Abschnitt 3.5.1.

13.1.1.4 Identifizierte Umgebung*

In einem großen Projekt ist es wichtig, die Anzahl und die Arten der benötigten Umgebungen zu identifizieren. Es werden mehrere sein, da zahlreiche Scrum Teams ihre Sprints am selben Tag beginnen und abschließen. Einige Beispiele für solche Umgebungen sind: Software-Entwicklungs-/Test-Bereich, physische Arbeitsbereiche/Ressourcen oder Prozessgrenzen für jedes Scrum Team.

13.1.1.5 Empfehlungen des Scrum Guidance Body

Beschrieben in Abschnitt 8.1.1.11.

Für große Projekte ist der Scrum Guidance Body eine wichtige Quelle für Empfehlungen und Anleitungen und zur Verbreitung von Besten Praktiken zur Verbesserung der Erfolgchancen des Projekts.

13.1.1.6 Product Owner

Die Rolle eines Product Owners ist im Großen und Ganzen dieselbe für kleine und große Projekte und ist in Abschnitt 3.4 beschrieben. Dies schließt die Kollaboration von Product Ownern mit ihren jeweiligen Scrum Teams ein.

Der wesentliche Unterschied in einem großen Projekt ist, dass der Product Owner die täglichen Prioritätsentscheidungen nicht mehr trifft, sondern nur dem Chief Product Owner Inputs liefert und Empfehlungen gibt. Darüber hinaus werden Interaktionen mit Stakeholdern unter allen Product Ownern aufgeteilt. Jeder Product Owner arbeitet mit bestimmtem Team (s) entsprechend definierten Rollen. All das wird im Product Owner Kollaborationsplan festgelegt.

Product Owner arbeiten mit dem Chief Scrum Master, Chief Product Owner, Scrum Mastern, und anderen Product Ownern zusammen, um die Liste der Komponenten und Ressourcen zu entwickeln, die von allen Teams im gesamten Projekt gemeinschaftlich benötigt werden. Sie helfen auch bei der Entwicklung und Verabschiedung des Release Vorbereitungs-Plans, der z.B. einen abschließenden End-to-End Integrationstest einschließen könnte.

13.1.1.7 Scrum Master

Die Rolle eines Scrum Masters ist im Großen und Ganzen dieselbe für kleine und große Projekte und ist in Abschnitt 3,5 beschrieben.

Ein großes Projekt hat im Allgemeinen mehrere Scrum Master - jeder von ihnen unterstützt und stellt ein produktives Arbeitsumfeld für sein / ihr jeweiliges Scrum Team sicher, schließt sich an - ein Scrum Master könnte mit mehr als einem Scrum Team arbeiten. Die Identifizierung der Scrum Master erfolgt im *Scrum Master- und Stakeholder- (s) Identifizieren* Prozess (siehe Abschnitt 8.2).

Die Zusammenarbeit zwischen den verschiedenen Scrum Teams, moderiert durch die Scrum Master, wird im Scrum Teams Kollaborationsplan festgelegt (siehe Abschnitt 13.1.3.8).

Scrum Master kollaborieren mit dem Chief Scrum Master, Chief Product Owner, anderen Scrum Mastern, und Product Ownern, um die Liste von Komponenten und Ressourcen zu entwickeln, die von allen Teams im gesamten Projekt in gemeinschaftlich benötigt werden. Sie helfen auch dabei, Inputs für die Entwicklung des Release Vorbereitungs-Plans zu liefern.

13.1.1.8 Unternehmensorganisation

Organisationen, die beabsichtigen, Scrum für große Projekte zu verwenden, sollten den Scrum-Rahmen vollständig annehmen. Für ein großes Projekt sollte die Organisation in der Lage sein, die notwendigen Aufwendungen durch Einsetzen der erforderlichen Ressourcen zu unterstützen. Wenn sie das nicht kann, müssen Pläne gemacht werden, um mehr Ressourcen (Mitarbeiter, Hilfsmittel und Arbeitsraum) zu bekommen. Es ist unbedingt nötig, dass ein Unternehmen, das plant, Scrum zu verwenden, darauf vorbereitet ist, seine Arbeitskultur und seine Gewohnheiten anzupassen, um den vollen Nutzen der Anwendung von Scrum zu erzielen.

13.1.1.9 **Business Case**

Beschrieben in Abschnitten 4.4.1, 4.4.2 und 8.1.1.1.

Obwohl der Business Case als solches nicht anders ist als ein Business Case für ein kleines Projekt, ist eine solide Geschäftsrechtfertigung mit der Zunahme der Größe eines Projekts zunehmend wichtig. Die Geschäftsauswirkung einer guten, und mehr noch einer schlechten Projektauswahl, sind bedingt durch den größeren Einsatz von Geld und anderen Ressourcen, die mit dem Projekt verbunden sind, bedeutend höher.

13.1.1.10 **Programm Scrum Master**

Beschrieben in Abschnitt 3.5.2.

13.1.1.11 **Programm Product Owner**

Beschrieben in Abschnitt 3.4.3.

13.1.1.12 **Organisatorische Ressourcenmatrix**

In einer Umgebung eines großen Scrum Projekts mit zahlreichen Scrum Teams, die bezüglich Zuweisung von Ressourcen und Priorisierung von Aufgaben miteinander konkurrieren, ist es wichtig, die organisatorischen Ressourcen auf eine optimale Weise zu verwalten, um die allgemeinen Projektziele zu erreichen. Product Owner müssen Wertschaffung dadurch handhaben, dass sie organisatorische Pläne zur Hand haben. Die organisatorischen Pläne sollten die zu entwickelnden Komponenten sowie die Fertigkeiten, Kosten und andere Ressourcen beinhalten, die zur Entwicklung dieser Komponenten benötigt werden, außerdem die aktuellen Geschwindigkeiten der einzelnen Scrum Teams (um die voraussichtliche Dauer des Projekts grob abzuschätzen), Kommunikationsanforderungen und andere Schnittstellen, die das Scrum pflegen muss.

Beschrieben in Abschnitt 8.2.1.10.

13.1.2 Hilfsmittel

13.1.2.1 Umgebung Planungs Meeting*

Ein Umgebung Planungs Meeting wird verwendet, um einen Zeitplan/Kalender auszuarbeiten, der beschreibt, wie die Scrum Teams Umgebungen untereinander teilen. Zum Beispiel kann es mit verteilten Teams, die in verschiedenen Zeitzonen arbeiten, möglich sein, rund um die Uhr Tests durchzuführen, um die Verwendung der vorhandenen Umgebungen zu maximieren. Dementsprechend ist es notwendig, einen Plan zu erstellen, der die Testzeiten für jedes Team festlegt. Für Softwareprojekte kann der Umgebungsplan auch Informationen beinhalten, wie und von wem der Code in jede Umgebung eingebracht wird.

13.1.2.2 Kommunikationsplan

Kommunikationspläne sind in einem großen Projekt sehr wichtig, da Missverständnisse oder Mangel an Kommunikation zwischen den zahlreichen Scrum Teams den gemeinsamen Anstrengungen abträglich sind und zu einem Misserfolg des Projekts führen können. Kommunikationspläne sollten wesentliche Informationen identifizieren sowie die Kommunikationsmethoden, Kanäle oder Mechanismen zur Übermittlung dieser Informationen an die richtigen Stakeholder definieren. Dazu gehört z. B. wie der Chief Product Owner und die anderen Product Owner Prioritäten an die Scrum Teams weitergeben, wer für Kommunikation verantwortlich ist, die Klassifizierung vertraulicher Informationen, das Timing von Kommunikationsaktivitäten sowie der Prozess, die Wirksamkeit der Kommunikation zu beurteilen. Die Kommunikationspläne sollten auch das Timing und die Häufigkeit von Scrum der Scrum Meetings (SOS) sowie ihren Durchführungs-Modus einschließen.

Jedes Scrum Team kann einen eigenen Kommunikationsplan haben, der festlegt, welche Aufzeichnungen für das Projekt erstellt und gepflegt werden sollen, sowie die Art und Weise, wie wichtige Informationen an die Stakeholder weitergegeben werden. Der team-spezifische Kommunikationsplan definiert auch, wer für verschiedene Kommunikationsaktivitäten verantwortlich ist. Siehe Abschnitt 12.1.2.2.

13.1.2.3 Ressourcenplanung für das Große Projekt

Ressourcenplanung für das Große Projekt ist aufgrund der Komplexität, verschiedene Arten von Ressourcen an die zahlreichen, parallel arbeitender Scrum Teams zu vergeben, sehr wichtig. Es gibt miteinander konkurrierenden Bedarf für knappe Ressourcen, und der Chief Product Owner sowie die anderen Product Owner müssen planen, wie sie in kürzester Zeit den größten Wert liefern können. Ressourcenplanung in einem großen Projekt sollte die verschiedenen mit Ressourcen wie Personal, dessen Ausbildung, Hardware und Software, externen Diensten und physischem Arbeitsraum verbundenen Kosten in Betracht ziehen.

Der Chief Product Owner und die anderen Product Owner müssen möglicherweise mit externen Quellen koordinieren, um zusätzliche Ressourcen zu akquirieren und so den Mitarbeiterstab zu erweitern, (es mag z.B. notwendig sein, externe Ressourcen zur Zusammenarbeit mit dem vorhandenen Ganztags-Team zu akquirieren. Das erfolgt gegebenenfalls in Zusammenarbeit mit dem firmeninternen Partner-Management-Team). Bei der Akquirierung von externen Ressourcen, müssen der Chief Product Owner und das Team Unternehmensrichtlinien für den Umgang mit externen Ressourcen und Anbietern einhalten.

In großen Projekten muss der Chief Product Owner möglicherweise eine zusätzliche Ressourcenplanung durchführen, um den Bedarf an spezialisierten Teams sowie den Bedarf nach Arbeitsumgebungen für die zahlreichen parallel arbeitenden Scrum Teams zu adressieren. Der Chief Product Owner und die Product Owner arbeiten normalerweise mit Scrum Mastern und Scrum Teams zusammen, um die speziellen, für das große Projekt erforderlichen Fertigkeiten, die Anzahl der erforderlichen Ressourcen, die benötigten speziellen Fertigkeiten der einzelnen Scrum Teams sowie eine entsprechende Zuordnung zu ermitteln bzw. abzuschätzen.

13.1.2.4 **Abhängigkeitsermittlung**

Beschrieben in Abschnitt 9.4.2.3.

Abhängigkeiten korrekt zu ermitteln hilft in großen Projekten den Scrum Teams, festzustellen, welche ihrer Entscheidungen andere Teams betreffen könnten. Die Abhängigkeiten können auch die Reihenfolge, in der ein einzelnes Scrum Team seine jeweiligen Aufgaben zur Schaffung der zu liefernden Ergebnisse des Sprints ausführt, beeinflussen.

13.1.3 **Outputs**

13.1.3.1 **Release Vorbereitungsplan***

Aufgrund der Tatsache, dass jeder Sprint ein potentiell lieferbares Produkt oder anderes zu lieferndes Ergebnis erzeugt, kann in einem kleinen Projekt ein Release nach jedwedem Sprint erfolgen, sofern es geschäftlich Sinn macht. In einem großen Projekt kann es vernünftige geschäftliche Gründe geben, nicht in jedem Sprint alle Aktivitäten durchzuführen, die notwendig sind, um für ein Release wirklich bereit zu sein. Zum Beispiel kann ein Projektteam beschließen, eine vollständige Menge von teuren und zeitraubenden Tests der Leistungsfähigkeit eines Produkts oder einen speziellen End-to-End Tests nur zeitnah zu einer Freigabe durchzuführen. Die festgelegten Done Kriterien für regelmäßige Sprints muss dennoch erfüllt werden.

Falls solche geschäftlich begründeten Entscheidungen getroffen werden, ergibt sich die Notwendigkeit für einen speziellen Release-Vorbereitungs-Sprint. (siehe 13.3 Ein Release im Großen Projekt Vorbereiten).

Diese geschäftlich begründeten Entscheidungen mit der zugehörigen Geschäftlichen Rechtfertigung werden in einem Release Vorbereitungsplan beschrieben. Der Release Vorbereitungsplan stellt ausführlich dar, welche Schritte von jedem Scrum Team sowie dem Projekt als Ganzes durchgeführt werden müssen, um zu bestätigen, dass die Mindestanforderungen für Freigabe erfüllt sind, und das Produkt oder eine seiner Komponenten bereit zur Freigabe ist.

13.1.3.2 Minimale Done Kriterien

Beschrieben in Abschnitt 5.4.3. Done Kriterien als solche sind in Abschnitt 8.5.3.2 beschrieben.

Die Done Kriterien für ein großes Projekt sollten als "Minimale" Done Kriterien angesehen werden. Product Owner können zusätzliche Done Kriterien für bestimmte Arten von User Storys definieren (z.B. für alle User Storys, die zur Schaffung oder Änderungen einer Website führen). Falls solche zusätzlichen Done Kriterien definiert sind, müssen sie von den jeweiligen Scrum Teams ebenfalls berücksichtigt werden. Done Kriterien können als Checklisten definiert werden, die bei Sprints und bei den Releases verwendet werden können. Einzelne Product Owner und ihre Scrum Teams können die zusätzlichen Done Kriterien definieren, die sie als notwendig oder wünschenswert erachten.

13.1.3.3 User Story Abnahmekriterien

Beschrieben in Abschnitt 9.1.3.2.

13.1.3.4 Geteilte Ressourcen

Geteilte Ressourcen können Personal, Umgebung und Ausrüstung einschließen, die von allen oder mehreren Scrum Teams im Projekt benötigt werden. In einem großen Projekt können die geteilten Ressourcen beschränkt sein und/oder von allen oder mehreren Scrum Teams zur gleichen Zeit benötigt werden.

Vor diesem Hintergrund müssen der Chief Product Owner, der Chief Scrum Master, andere Product Owner und andere Scrum Master eine Methode zur Verwendung dieser geteilten Ressourcen entwickeln. In einem großen Projekt ist die Verantwortung, die zu liefernden Ergebnisse für einen Sprint zu liefern, auf zahlreiche Scrum Teams verteilt, die jeweils unterschiedliche Prioritäten und Verfügbarkeit haben. Eine mögliche Vorgehensweise für die Zuweisung von geteilten Ressourcen könnte sein, sicherzustellen, dass Ressourcen zuerst für die wichtigsten / höchstwertigen Features und an die an diesen arbeitenden Teams vergeben werden. Wenn konkurrierende Anforderungen eine vergleichbare Priorität haben, sollte der Chief Product Owner die Entscheidung über die Zuweisung basierend auf den neusten geschäftlichen Anforderungen treffen.

13.1.3.5 Team Spezialisierung

In einem großen Projekt kann es sein, dass eine Spezialisierung von Teams erforderlich ist. Es gibt drei Dimensionen der Team Spezialisierung.

Die erste Dimension ist der Bedarf danach, bestimmte Aufgaben zu erledigen. Zum Beispiel könnte ein spezialisiertes Team ein Integrationsteam sein, das spezielle Kenntnisse von fortlaufender Integration (Continuous Integration) hat. Diese Kenntnisse können besonders wichtig sein, wenn und falls es einen Release Vorbereitungs-Sprint gibt.

Die zweite Dimension ist der Bedarf nach speziellen Fertigkeiten von einzelnen Teammitgliedern. Theoretisch sind alle Scrum Team Mitglieder Generalisten und Spezialisten, d.h. sie haben Kenntnis von verschiedenen Bereichen und sind Experten in mindestens einem Bereich. In einem großen Projekt kann das allerdings anders sein. Mitglieder von spezialisierten Teams müssen möglicherweise bestimmte Fertigkeiten besitzen, - zum Beispiel spezielles Domänenwissen wie Sicherheit- die nicht unbedingt in allen Teams des großen Projekts verfügbar sind, für die sie erforderlich wären. Für ein großes Projekt wäre es äußerst kostspielig, alle Mitglieder jedes einzelnen Teams in allen Domänen auszubilden. Experten mit solchen speziellen Fertigkeiten und Wissen können als temporäre Teammitglieder in verschiedenen Teams arbeiten. Manchmal mag es notwendig sein, sie von externen Quellen zu akquirieren. Ein neues Teammitglied hinzuzufügen, wird die Teamgeschwindigkeit beeinflussen.

Die dritte Dimension ist eine potentielle Einschränkung hinsichtlich der Flexibilität von Teams. Wie oben erwähnt, wäre es in einem großen Projekt äußerst kostspielig, jedes Teammitglied in allen Domänen auszubilden. Das bedeutet auch, dass jedes Team eine oder mehrere Domänen hat, in denen es gut ist, sowie einige Domänen, an denen es mit zusätzlichem Training und weiteren Inputs arbeiten kann. Es wird auch Domänen geben, an denen es nicht sinnvoll arbeiten kann. Das soeben gesagte kommt bei der Sprint Planung zum Tragen. Jedes Team eine Teilmenge von User Storys, die natürlicherweise im Sprint von ihm bearbeitet werden wird, einige User Stories, an denen es im Sprint sinnvoll arbeiten kann, obwohl diese nicht zu seiner Kernkompetenz gehören, und manche, die es nicht übernehmen kann, weil es nicht das notwendige Wissen / die notwendigen Fertigkeiten hat.

Das Projekt muss hier das Risiko eingehen, dass aufgrund dieser Beschränkungen in einem Sprint nicht alle der höchst-prioren User Storys adressiert werden können. Dementsprechend kann es notwendig werden, einige User Stories von geringerer Priorität zu entwickeln, um den spezialisierten Fertigkeiten von noch nicht bereits ausgebuchten Teams Rechnung zu tragen.

13.1.3.6 Empfohlene Verbesserungen des Scrum Guidance Body

In Folge der Planung für das große Projekt können Vorschläge gemacht werden, die Empfehlungen des Scrum Guidance Body zu überarbeiten oder zu verbessern. Wenn der Guidance Body diese Vorschläge akzeptiert, werden sie als Updates in die Scrum Guidance Body Dokumentation integriert.

13.1.3.7 Product Owner Kollaborationsplan

Der Product Owner Kollaborationsplan sollte definieren, wie mehrere Product Owner mit dem Chief Product Owner zusammenarbeiten. Er sollte mindestens definieren, mit wie vielen Scrum Teams ein einzelner Product Owner direkt zusammenarbeiten kann (auf Grundlage von Erfahrung, Zeit, Domänenwissen), wie die Sammlung von Anforderungen von den Stakeholdern unter den Product Ownern aufgeteilt wird, wie das Priorisierte Produkt Backlog mit neuen Anforderungen oder Änderungen in bestehenden Anforderungen aktualisiert wird, wie die Zusammenarbeit der Product Owner mit mehreren Scrum Teams aussieht. Es sollte beachtet werden, dass jedes Scrum Team nur einen einzigen Product Owner als direkten Ansprechpartner hat. Jedoch kann ein einzelner Product Owner mit mehr als einem Scrum Team direkt zusammenarbeiten, falls erforderlich.

13.1.3.8 Scrum Teams Kollaborationsplan

Der Scrum Teams Kollaborationsplan definiert, wie die zahlreichen Scrum Teams zusammenarbeiten, um in der kürzesten Zeit den höchsten möglichen Wert zu liefern. Dieser Plan sollte sowohl Information über die Zuordnung von speziellen Domänen zu qualifizierten Teams beinhalten als auch, wie die Teams die Pflege des Priorisierten Produkt Backlogs und Abschätzungen unterstützen (d.h. die Festlegung, welche der Teammitglieder an Meetings zur Pflege des Product Backlogs oder zur High-Level Aufwandsabschätzung teilnehmen) und wie die Teams die Scrum der Scrums (SOS) Meetings organisieren.

Dieser Plan kann auch Information darüber einschließen, welches Coaching jedes Scrum Team erhält (z.B. ob es einen separaten Trainer zusätzlich zu seinem Scrum Master gibt; ob es im Falle von verteilten Teams einen Scrum Master an jedem Standort gibt; wie die Teammitglieder mit Scrum Mastern am selben Standort und mit Scrum Mastern an anderen Standorten zusammenarbeiten).

13.1.3.9 Abhängigkeiten

Beschrieben in Abschnitt 9.4.3.3.

13.2 Sprints Durchführen und Koordinieren

Abbildung 13-5 zeigt alle Inputs, Hilfsmittel und Outputs für den *Sprints Durchführen und Koordinieren* Prozess.

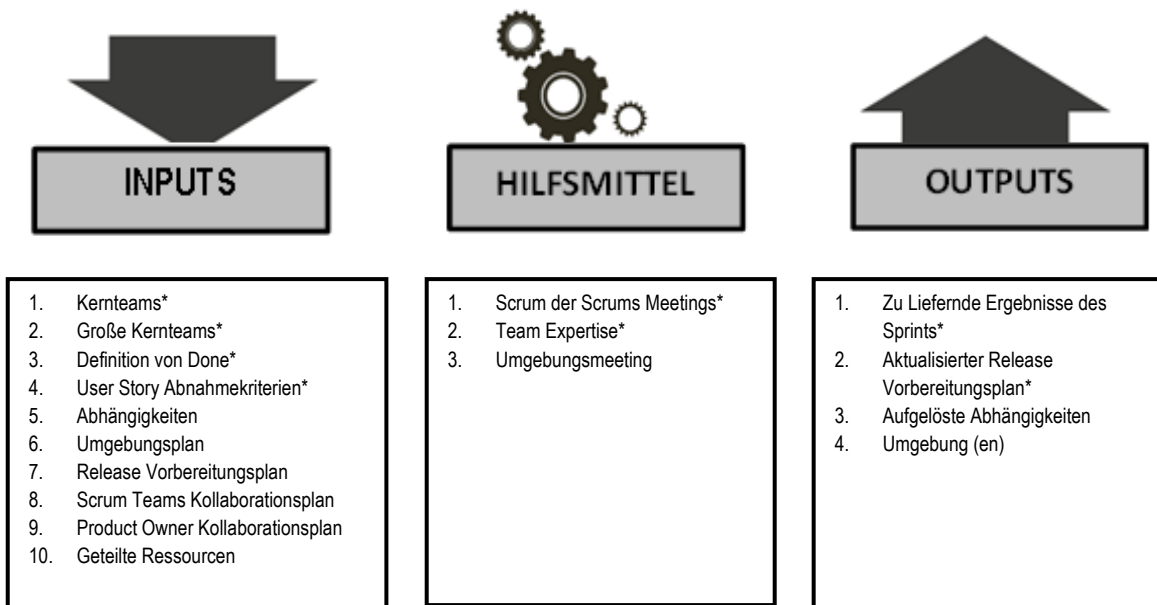


Abbildung 13-5: Sprints Durchführen und Koordinieren - Inputs, Hilfsmittel und Outputs

Hinweis - Sternchen (*) bezeichnen einen "verbindlichen" Input, ein "verbindliches" Hilfsmittel oder einen "verbindlichen" Output für den entsprechenden Prozess.

Abbildung 13-6 zeigt alle Beziehungen des *Programm- oder Portfolio- Komponenten Erstellen* Prozesses zu den grundlegenden Scrum Prozessen.

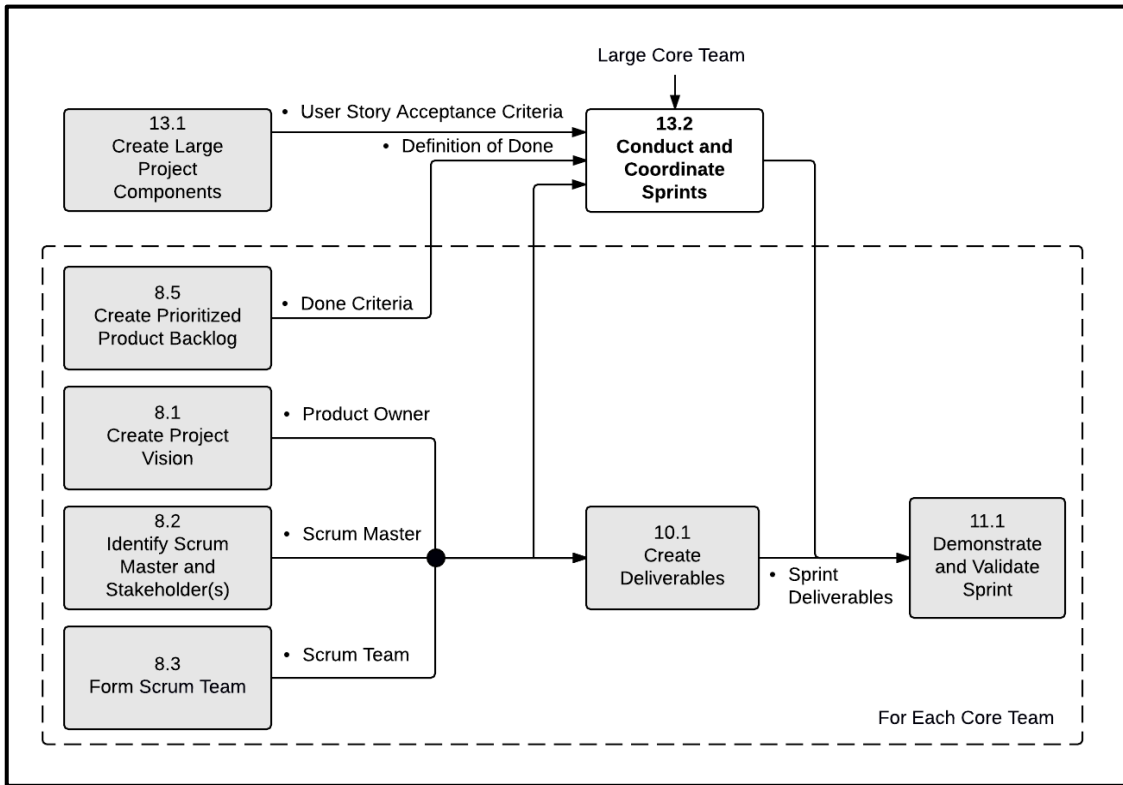


Abbildung 13-6: Sprints Durchführen und Koordinieren - Datenflussdiagramm

13.2.1 Inputs

13.2.1.1 Kernteams*

Die zahlreichen aus Scrum Master, Product Owner, und Scrum Team bestehenden Kernteams sind in Abschnitten 8.4.1.1 und 3.3.1 beschrieben.

13.2.1.2 Großes Kernteam*

Das große Kernteam umfasst den Chief Product Owner, den Chief Scrum Master, die Scrum Master, die Product Owner und ausgewählte Mitglieder der Scrum Teams, die am großen Projekt arbeiten. Ein großes Kernteam zu haben, das alle Mitglieder aller Teams einschließt, wäre unpraktisch. Deshalb können Teams ein Mitglied wählen, das sie im großen Kernteam vertreten sollte.

Der Chief Scrum Master beseitigt Hindernisse und stellt eine produktive Projektumgebung für alle Scrum Teams, die am Projekt beteiligt sind, sicher. Der Chief Product Owner bereitet das Priorisierte Produkt Backlog für das große Projekt vor und pflegt es, um mit seiner Hilfe die Arbeit durch die Product Owner der Scrum Teams zu koordinieren. Product Owner sind dafür verantwortlich, die Priorität von Features und Komponenten, die von den einzelnen am großen Projekt beteiligten Scrum Teams zu entwickeln sind, festzulegen.

Chief Scrum Master, Chief Product Owner, Scrum Master, Product Owner, und die ausgewählten Mitglieder der Scrum Teams arbeiten zusammen, um eine Liste der Komponenten und Ressourcen zu entwickeln, die von allen Teams des Projekts gemeinsam benötigt werden.

13.2.1.3 Definition von Done*

Beschrieben in Abschnitten 8.5.3.2 und 5.4.3.

Falls es einen Release Vorbereitungs-Sprint gibt, können und werden seine Done Kriterien wahrscheinlich anders sein als die von anderen Sprints. Done Kriterien werden definiert, um sicherzustellen, dass die zu Liefernden Ergebnisse des Sprints "potenziell versandfähig" sind. Der Release Vorbereitungs-Sprint adressiert alle Angelegenheiten, die in regelmäßigen Sprints wegen einer bewussten Geschäftlichen Entscheidung nicht adressiert wurden, und die im Release Vorbereitungsplan aufgezeichnet sind.

13.2.1.4 User Story Abnahmekriterien*

Beschrieben in Abschnitt 9.1.3.2.

13.2.1.5 **Abhängigkeiten**

Beschrieben in Abschnitten 9.3.2.4 und 9.3.3.3.

13.2.1.6 **Umgebungsplan**

Der Umgebungsplan ist ein Plan/Kalender, der beschreibt, wie die Scrum Teams die Umgebungen untereinander teilen. Er beschreibt die jedem Team zugeteilten Tage und Zeitperioden, an denen es jede Umgebung verwenden kann.

13.2.1.7 **Release Vorbereitungsplan**

Beschrieben in Abschnitt 13.1.3.1.

13.2.1.8 **Scrum Teams Kollaborationsplan**

Beschrieben in Abschnitt 13.1.3.8.

13.2.1.9 **Product Owner Kollaborationsplan**

Beschrieben in Abschnitt 13.1.3.7.

13.2.1.10 **Geteilte Ressourcen**

Geteilte Ressourcen können Personal, Umgebung und Ausrüstung einschließen, die von allen oder mehreren Scrum Teams im Projekt benötigt werden. In einem großen Projekt können die geteilten Ressourcen beschränkt sein und/oder von allen oder mehreren Scrum Teams zur gleichen Zeit benötigt werden.

Siehe Abschnitt 13.1.3.4.

13.2.2 Hilfsmittel

13.2.2.1 Scrum der Scrums Meeting*

Ein Scrum der Scrums Meeting ist ein wichtiges Element, wenn Scrum auf ein Großprojekt zugeschnitten wird. Üblicherweise ist ein Vertreter von jedem Scrum Team im Meeting anwesend—normalerweise der Scrum Master—aber es ist auch verbreitet, dass falls nötig, irgendjemand aus dem Team an dem Meeting teilnimmt. Das Meeting wird für in der Regel vom Chief Scrum Master organisiert und dient dem Fokus auf Koordinations- und Integrationsbereiche zwischen den einzelnen Scrum Teams.

Diese sind vorzugsweise kurze Treffen (aber in der Regel nicht Zeit-begrenzt, um mehr Informationsaustausch zwischen den Teams zu ermöglichen), bei denen sich Vertreter aus jedem Scrum Team treffen, um sich über den Status der jeweiligen Teams auszutauschen. Das Scrum der Scrums Meeting wird in vorherbestimmten Abständen, oder wenn sie von einem Scrum Team benötigt werden, abgehalten. Angelegenheiten, Abhängigkeiten und Risiken, welche mehrere Scrum Teams betreffen, können genau beobachtet werden, was den verschiedenen Teams dabei hilft, die Arbeit an einem großen Projekt besser zu koordinieren und ihre Arbeit zu integrieren. Es ist die Aufgabe des Chief Scrum Master (oder eines anderen Scrum Masters, der das Scrum der Scrums Meeting leitet), sicherzustellen, dass alle Vertreter in einem Umfeld agieren, dass einen offenen und ehrlichen Austausch von Informationen, einschließlich von Feedback an andere Teamvertreter, fördert. Für größere Projekte, mit einer umfangreichen Anzahl von Teams, kann diese Art von Treffen auf mehreren Ebenen abgehalten werden.

Einer nach dem andern liefern die Vertreter der Scrum Teams Updates von ihrem jeweiligen Team. Diese Updates werden in der Regel in Form von Antworten auf vier konkrete Fragen geliefert.

1. Woran hat mein Team seit dem letzten Treffen gearbeitet?
2. Woran wird mein Team bis zum nächsten Treffen arbeiten?
3. Welche der Aufgaben, von denen andere Teams erwartet haben, dass mein Team sie löst, sind noch nicht gelöst?
4. Welche Aktivitäten, die andere Teams beeinflussen können, werden von meinem Team geplant?

Die Antworten auf diese vier Fragen liefern Informationen, die jedes Team in die Lage versetzen, den Arbeits- Status aller anderen Teams klar zu verstehen.

13.2.2.2 Team Fachkenntnisse*

Beschrieben in Abschnitt 10.1.2.1.

Für große Projekte schließt dies auch das Wissen des Chief Product Owners und Chief Scrum Masters über die Teamfachkenntnis ein. Letzteres ist hinsichtlich einer Bewertung der Optionen für die Entwicklung der User Stories der einzelnen Sprints hilfreich.

13.2.2.3 Umgebungsmeeting

Dieses Meeting wird abgehalten, um die Arten und die Anzahl der Umgebungen zu ermitteln, die notwendig sind, um die zu liefernden Ergebnisse des Projekts zu entwickeln, zu verwalten und zu testen. In diesem Meeting werden auch die Ressourcen erörtert, die benötigt werden, um die erforderlichen Umgebungen einzuführen.

13.2.3 Outputs

13.2.3.1 Zu Liefernde Ergebnisse des Sprints*

Am Ende eines jeden Sprints ist ein Produktinkrement oder zu lieferndes Ergebnis fertiggestellt, das potentiell versandfähig ist. Die Lieferung sollte alle definierten Eigenschaften und Funktionalitäten, welche in den User Stories des Sprints definiert sind, enthalten. Sie sollten auch erfolgreich getestet worden sein.

13.2.3.2 Aktualisierter Release Vorbereitungsplan*

Der Release Vorbereitungsplan detailliert die Schritte, die jedes Scrum Team und das Projekt als Ganzes ausführen müssen, um zu bestätigen, dass die Mindestanforderungen für das Release erfüllt sind. Er kann mit in diesem Prozess identifizierten Änderungen aktualisiert werden.

13.2.3.3 Aufgelöste Abhängigkeiten

Abhängigkeiten zwischen User Storys, Aufgaben und Ressourcen, die benötigt werden, User Storys in zu Liefernde Ergebnisse zu verwandeln, machen es notwendig, dass ein großes Projekt einen Aktionsplan zur Auflösung von Abhängigkeiten hat. Dieser Plan deckt alle Aktionen, die notwendig sind, um alle Arten von Abhängigkeiten zu handhaben, - zwingende, nicht-zwingende, externe und interne Abhängigkeiten.

13.2.3.4 Umgebung (en)

Dies bezieht sich auf die Identifikation und Dokumentation aller Umgebungen, die benötigt werden, um die zu liefernden Ergebnisse des Projekts zu entwickeln und zu testen.

13.3 Ein Release im Großen Projekt Vorbereiten

Abbildung 13-7 zeigt alle Inputs, Hilfsmittel und Outputs für den *Ein Release im Großen Projekt Vorbereiten* Prozess.



Abbildung 13-7: Ein Release im Großen Projekt Vorbereiten - Inputs, Hilfsmittel und Outputs

Abbildung 13-8 zeigt alle Beziehungen des *Ein Release im Großen Projekt Vorbereiten* Prozesses zu den grundlegenden Scrum Prozessen.

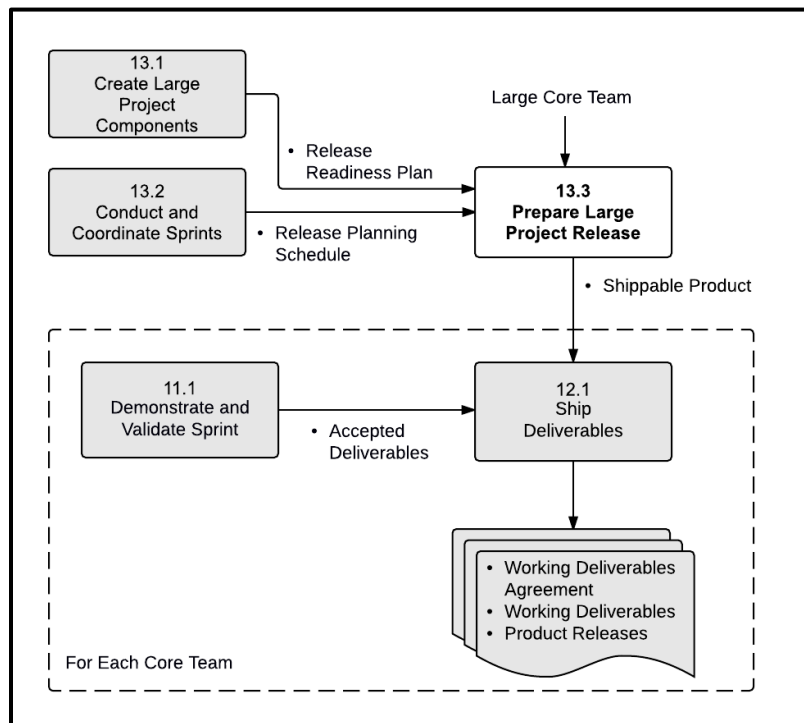


Abbildung 13-8: Ein Release im Großen Projekt Vorbereiten - Datenflussdiagramm

Hinweis - Sternchen (*) bezeichnen einen "verbindlichen" Input, ein "verbindliches" Hilfsmittel oder einen "verbindlichen" Output für den entsprechenden Prozess.

13.3.1 Inputs

13.3.1.1 Kernteams*

Beschrieben in Abschnitten 8.4.1.1 und 3.3.1.

13.3.1.2 Großes Kernteam*

Beschrieben in Abschnitt 13.2.1.2.

13.3.1.3 Release Plan*

Beschrieben in Abschnitt 8.6.3.1.

13.3.1.4 Release Vorbereitungsplan

Beschrieben in Abschnitt 13.1.3.1.

13.3.2 Hilfsmittel

13.3.2.1 Kommunikationsplan*

Beschrieben in Abschnitten 12.1.2.2 und 13.1.2.1.

13.3.2.2 Release Vorbereitungs-Sprint

Wenn es einen Bedarf gibt, bestimmte Aufgaben zur Vorbereitung zu einem Release auszuführen und zu bestätigen, dass den Mindestanforderungen für ein Release entsprochen wurde, werde diese Aufgaben in einem Release Vorbereitungs-Sprint ausgeführt. Einen Release Vorbereitungs-Sprint, falls erforderlich, gibt es nur einmal pro Release, und zwar als den letzten Sprint vor der Freigabe. In einem Release Vorbereitungs-Sprint werden keine User Storys aus dem Priorisierten Produkt Backlog entwickelt.

Stattdessen werden die Aufgaben ausgeführt, die im Release Vorbereitungsplan identifiziert sind (siehe 13.1.3.1).

Es ist wichtig, nicht zu vergessen: diese Aufgaben werden in einem Release Vorbereitungs-Sprint infolge einer geschäftlichen Entscheidung ausgeführt. Das ändert Nichts an der Vorgabe, am Ende jedes regulären Sprints alle Abnahmekriterien und Done Kriterien zu erfüllen.

Ein Release Vorbereitungs-Sprint ist nicht zwingend vorgeschrieben. Obwohl er nicht zwingend vorgeschrieben ist, wird in vielen großen Projekten eine geschäftliche Entscheidung getroffen, einen Release Vorbereitungs-Sprint zu haben.

13.3.2.3 Release Vorbereitungsmethoden

Release Vorbereitungsmethoden sind die Methoden, die angewendet werden, um die Aufgaben auszuführen, die im Release Vorbereitungsplan identifiziert wurden, um die zu liefernden Ergebnisse auslieferungs-/ freigabe-fertig zu machen. Diese Aufgaben können projektspezifisch sein, sind aber eher auf Portfolio- oder mindestens Programmebene gültig. Sie können im Scrum Guidance Body definiert werden.

13.3.3 Outputs

13.3.3.1 Lieferbares Produkt*

Ein Lieferbares Produkt ist ein zu Lieferndes Ergebnis oder ein Produktinkrement, das die durch den Kunden und den Chief Product Owner definierten Abnahmekriterien entspricht. Es ist am Ende eines Release vorbereitungs-Sprints zur auslieferungs- oder freigabe-fertig.

13.3.3.2 Freigabemitteilung

Die Freigabemitteilung beinhaltet Dokumentation, die dem Kunden zusammen mit dem Produktrelease geliefert wird. Sie beinhaltet externe oder marktorientierte Lieferkriterien für das zu liefernde Produkt.

13.3.3.3 Release Umgebung

Die Release Umgebung wird benötigt, um das Release im Einsatz zu unterstützen

13.3.3.4 Empfohlene Verbesserungen des Scrum Guidance Body

Beschrieben in Abschnitt 13.1.3.6.

13.4 Die Auswirkung von großen Projekten auf Grundlegende Scrum Prozesse

Obwohl die grundlegenden Scrum Prozesse, beschrieben in Kapiteln 8 bis 12, für große Projekte gültig bleiben, gibt es bestimmte zu erwartende Auswirkungen, auf die hingewiesen werden sollte. Tabelle 13-1 fasst die Auswirkungen eines großen Projekts auf die grundlegenden Scrum Prozesse zusammen.

Prozess	Beschreibung der Auswirkungen auf existierende Prozesse für ein großes Projekt
8.1 Projektvision Erstellen	<p>Zusätzlicher Output: Identifizierter Chief Product Owner* Für ein großes Projekt wird ein Chief Product Owner als Output dieses Prozesses identifiziert. Während kleine Projekte nur einen Product Owner erfordern, werden für große Projekte mehrere Product Owner identifiziert.</p> <p>Zusätzlicher Output: Identifizierter Chief Scrum Master* Dem Chief Product Owner ähnlich, sollte auch der Chief Scrum Master für ein großes Projekt identifiziert werden.</p>
8.2 Scrum Master und Stakeholder identifizieren	<p>Output: Identifizierte Stakeholder Zusätzlich zur Identifikation von Stakeholdern des Projekts sollten für ein großes Projekt auch einige besonders wichtige Mitglieder von Scrum Teams in diesem Prozess als relevante Stakeholder identifiziert werden. Diese Stakeholder werden Inputs in den <i>Komponenten des Großen Projekts Erstellen</i> Prozess sein.</p>

Prozess	Beschreibung der Auswirkungen auf existierende Prozesse für ein großes Projekt
8.3 Scrum Team Bilden	<p>Zusätzlicher Input: Chief Product Owner</p> <p>Beschrieben in Abschnitt 3.4.2. Für ein großes Projekt ist der Chief Product Owner daran beteiligt, die Scrum Teams zu bilden und die Mitglieder auszuwählen. Der Chief Product Owner dient dabei den Interessen des großen Projekts als Ganzes, während "normale" Product Owner sich auf die einzelnen Teams fokussieren.</p> <p>Zusätzlicher Input: Chief Scrum Master</p> <p>Beschrieben in Abschnitt 3.5.1. Für ein großes Projekt ist der Chief Scrum Master daran beteiligt, die Bildung des Scrum Teams zu bestimmen die Scrum Teams zu bilden und die Mitglieder auszuwählen. Der Chief Scrum Master dient dabei den Interessen des großen Projekts als Ganzes, während "normale" Scrum Master sich auf die einzelnen Teams fokussieren.</p> <p>Zusätzlicher Input: Geteilte Ressourcen</p> <p>Beschrieben in Abschnitt 13.1.3.4. Kenntnis von jedweden für die Scrum Teams verfügbaren geteilten Ressourcen ist ein notwendiger Input zur Bildung der einzelnen Scrum Teams.</p> <p>Zusätzlicher Input: Team Spezialisierung</p> <p>Beschrieben in Abschnitt 13.1.3.5.</p>
8.4 EPIC (s) entwickeln	<p>Zusätzlicher Input: Product Owner Kollaborationsplan</p> <p>Beschrieben in Abschnitt 13.1.3.7. Dies ist einer der zentralen Outputs des <i>Komponenten des Großen Projekts</i> Erstellen Prozesses. Er definiert, wie die verschiedenen Product Owner untereinander und mit dem Chief Product Owner zusammenarbeiten. Er gibt an, wie sie mit den Stakeholdern zum Sammeln von Anforderungen zusammenarbeiten, wie sie das Priorisierte Produkt Backlog aktualisieren, und wie sie mit den verschiedenen Scrum Teams zusammenarbeiten (für jedes Scrum Team gibt es weiterhin genau einen Product Owner, der direkt mit diesem arbeitet. Allerdings müssen die Product Owner entscheiden, wie sie die Scrum Teams untereinander aufteilen und mit wie vielen Scrum Teams jeder Product Owner arbeitet.)</p>

Prozess	Beschreibung der Auswirkungen auf existierende Prozesse für ein großes Projekt
8.5 Priorisiertes Produkt Backlog erstellen	<p>Zusätzlicher Input: Product Owner Kollaborationsplan</p> <p>Beschrieben in Abschnitt 13.1.3.7. Da der Product Owner Kollaborationsplan definiert, wie die Product Owner das Priorisierte Produkt Backlog aktualisieren, ist der Product Owner Kollaborationsplan ein wichtiger Input für diesen Prozess.</p> <p>Zusätzlicher Input: Abhängigkeiten</p> <p>Beschrieben in Abschnitt 13.1.3.7.</p>
8.6 Release-Planung durchführen	<p>Zusätzlicher Input: Chief Scrum Master</p> <p>Beschrieben in Abschnitt 3.5.1.</p> <p>Zusätzlicher Input: Release Vorbereitungsplan</p> <p>Beschrieben in Abschnitt 13.1.3.1. Der Release Vorbereitungsplan beinhaltet wichtige Informationen, die die allgemeine Release Planung betreffen.</p>
9.1 User Storys erstellen	<p>Zusätzlicher Input: Product Owner Kollaborationsplan</p> <p>Beschrieben in Abschnitt 13.1.3.7.</p> <p>Zusätzlicher Input: Scrum Teams Kollaborationsplan</p> <p>Beschrieben in Abschnitt 13.1.3.8.</p>
9.2 User Storys abschätzen	<p>Zusätzlicher Input: Product Owner Kollaborationsplan</p> <p>Beschrieben in Abschnitt 13.1.3.7.</p> <p>Zusätzlicher Input: Scrum Teams Kollaborationsplan</p> <p>Beschrieben in Abschnitt 13.1.3.8.</p>
9.3 User Storys zusagen	Keine Änderung
9.4 Aufgaben erstellen	Keine Änderung

Prozess	Beschreibung der Auswirkungen auf existierende Prozesse für ein großes Projekt
9.5 Aufgaben abschätzen	Keine Änderung
9.6 Sprint Backlog Erstellen	Keine Änderung
10.1 Zu liefernden Ergebnisse erzeugen	Keine Änderung
10.2 Tägliches Standup durchführen	Keine Änderung
10.3 Priorisiertes Produkt Backlog pflegen	<p>Zusätzlicher Input: Product Owner Kollaborationsplan Beschrieben in Abschnitt 13.1.3.7. Der Product Owner Kollaborationsplan definiert wie die Product Owner das Priorisierte Produkt Backlog aktualisieren.</p> <p>Zusätzlicher Input: Scrum Teams Kollaborationsplan Beschrieben in Abschnitt 13.1.3.8. Der Scrum Teams Kollaborationsplan definiert, wie die Teams an der Pflege des Priorisierten Product Backlogs teilnehmen. Dieser Plan definiert außerdem, welche Vertreter der einzelnen Teams an der Pflege des Backlogs teilnehmen und wie diese ausgewählt werden.</p> <p>Zusätzlicher Input: Team Spezialisierung Beschrieben in Abschnitt 13.1.3.5.</p> <p>Zusätzlicher Output: Aktualisierter Release Vorbereitungsplan Änderungen am Produkt Backlog, die während seiner Pflege eingebracht werden, können Auswirkungen auf den Release Vorbereitungsplan haben (siehe Abschnitt 13.1.3.1).</p>

Prozess	Beschreibung der Auswirkungen auf existierende Prozesse für ein großes Projekt
11.1 Sprint vorführen und validieren	Dieser Prozess wird für jedes einzelne Scrum Team getrennt durchgeführt. Für jedes Team genehmigt der jeweilige Product Owner User Storys, das kann allerdings wegen wechselseitiger Abhängigkeiten sehr komplex sein. Zu manchen Zeiten mag es sein, dass der Chief Product Owner an jedem einzelnen Sprint Review Meeting teilnimmt. Manchmal mag eine End-to-end Validierung notwendig sein, durch die möglicherweise bestimmte Dinge in der Bestätigung durch ein einzelnes Team als Done erscheinen, jedoch im Zuge der End-to-end Validierung die endgültigen Abnahmekriterien nicht erfüllen.
11.2 Sprint Rückblick halten	<p>Zusätzlicher Input: Product Owner Kollaborationsplan</p> <p>Beschrieben in Abschnitt 13.1.3.7. Für große Projekte kann die Pflege des Produkt Backlogs sehr schwierig sein und zu Problemen führen und Zeit und Aufwand vergeuden. Deshalb ist es empfehlenswert, die Pflege des Product Backlogs zu einem spezifischen Thema der Retrospektive zu machen. Ein spezifischer Aspekt der Backlog Pflege wäre das Zusammenwirken der verschiedenen Product Owner und ihre Zusammenarbeit mit den verschiedenen Teams zur Pflege des Backlogs.</p> <p>Zusätzlicher Input: Scrum Teams Kollaborationsplan</p> <p>Beschrieben in Abschnitt 13.1.3.8. Für große Projekte kann die Pflege des Produkt Backlogs sehr schwierig sein und zu Problemen führen und Zeit und Aufwand vergeuden. Deshalb ist es empfehlenswert, die Pflege des Product Backlogs zu einem spezifischen Thema der Retrospektive zu machen. Ein spezifischer Aspekt der Backlog Pflege wäre das Zusammenwirken der verschiedenen Scrum Teams und ihre Zusammenarbeit mit den verschiedenen Product Ownern zur Pflege des Backlogs.</p>
12.1 Zu liefernde Ergebnisse ausliefern	<p>Zusätzlicher Input: Chief Product Owner</p> <p>Beschrieben in Abschnitt 3.4.2.</p> <p>Zusätzlicher Input: Chief Scrum Master</p> <p>Beschrieben in Abschnitt 3.5.1.</p> <p>Zusätzlicher Input: Freigabemitteilung</p> <p>Für große Projekte wird die Freigabemitteilung aus dem <i>Ein Release des Großen Projekts Vorbereiten</i> Prozess ein Input in diesen Prozess.</p>

Prozess	Beschreibung der Auswirkungen auf existierende Prozesse für ein großes Projekt
12.2 Projekt Rückblick halten	Zusätzlicher Input: Chief Product Owner Beschrieben in Abschnitt 3.4.2. Zusätzlicher Input: Chief Scrum Master Beschrieben in Abschnitt 3.5.1.

Tabelle 13-1: Zusammenfassung der Auswirkungen von großen Projekten auf grundlegende Scrum Prozesse

14. Scrum für große Projekte Skalieren

Dieses Kapitel enthält die Prozesse, die sich mit der Skalierung von Scrum für das Unternehmen auseinandersetzen: Programm- oder Portfolio- Komponenten Erstellen, Scrum Guidance Body Reviewen und Aktualisieren, Programm oder Portfolio Backlog Erstellen und Pflegen, Programm- oder Portfolio Komponenten Koordinieren, und Programm- oder Portfolio Release Rückblick Halten.

Die Skalierung von Scrum für das Unternehmen, 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 Stakeholder 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 bestmögliche Anwendung des Scrum Rahmens zu ermöglichen, präsentiert dieses Kapitel Inputs, Hilfsmittel und Outputs für jeden Prozess als „verbindlich“ oder „optional.“ Inputs, Hilfsmittel und Outputs, die mit Sternchen (*) markiert sind, sind verbindlich, während solche ohne Sternchen optional sind.

Es wird empfohlen, dass Personen, welche sich erst mit dem Scrum Rahmen und seinen Prozessen vertraut machen, sich primär mit den verbindlichen Inputs, Hilfsmitteln und Outputs beschäftigen, während Scrum Praktiker mit größerer Erfahrung danach streben sollten, sich ein noch tieferes Wissen über die Informationen in diesem gesamten Kapitel anzueignen.

Die Abbildung 14-1 zeigt eine Übersicht über die Prozesse der Skalierung für das Unternehmen, welche sich folgendermaßen aufteilen:

14.1 Programm- oder Portfolio- Komponenten Erstellen - in diesem Prozess identifizieren Programm- oder Portfolio- Product Owner und relevante Stakeholder gemeinsame Komponenten und Ressourcen, die für das Programm oder Portfolio erforderlich sind. Die minimalen Done Kriterien werden definiert, und alle übrigen Stakeholder werden identifiziert.

14.2 Scrum Guidance Body Reviewen und Aktualisieren - in diesem Prozess werden die Empfehlungen des Scrum Guidance Body regelmäßig von den Mitgliedern des Scrum Guidance Bodys überprüft und, falls notwendig, aktualisiert. In diesem Prozess werden auch Änderungen in der Mitgliedschaft des Scrum Guidance Bodys gehandhabt.

14.3 Programm- oder Portfolio- Backlog Erstellen und Pflegen - in diesem Prozess wird das Programm oder Portfolio Backlog erstellt, aktualisiert und gepflegt. Empfehlungen für Verbesserungen der Empfehlungen des Scrum Guidance Body können gemacht werden. Dem Fortschritt entsprechend und/oder auf Grundlage von geänderten Anforderungen der Projekte des Programms oder Portfolios werden die Zieltermine für die Implementierung gegebenenfalls angepasst.

14.4 Programm- oder Portfolio- Komponenten Koordinieren - in diesem Prozess werden die Komponenten des Programms oder Portfolios koordiniert. Abhängigkeiten zwischen Projekten werden adressiert, gemeinsame Hindernisse werden erörtert, und Beste Praktiken werden ausgetauscht. Manchmal werden Empfehlungen für Verbesserungen des Scrum Guidance Bodys gemacht.

14.5 Programm oder Portfolio Releases Rückblick Halten - in diesem Prozess kommen der Programm oder Portfolio Product Owner und relevante Stakeholder zusammen, um auf ein Programm- oder Portfolio-Release zurückzublicken und Lehren aus dem Release zu identifizieren, zu dokumentieren und zu verinnerlichen. Oft führen diese Lehren zur Dokumentation von vereinbarten Maßnahmen, die in zukünftigen Releases umgesetzt werden. Manchmal werden Empfehlungen zur Verbesserung des Scrum Guidance Body ausgesprochen.

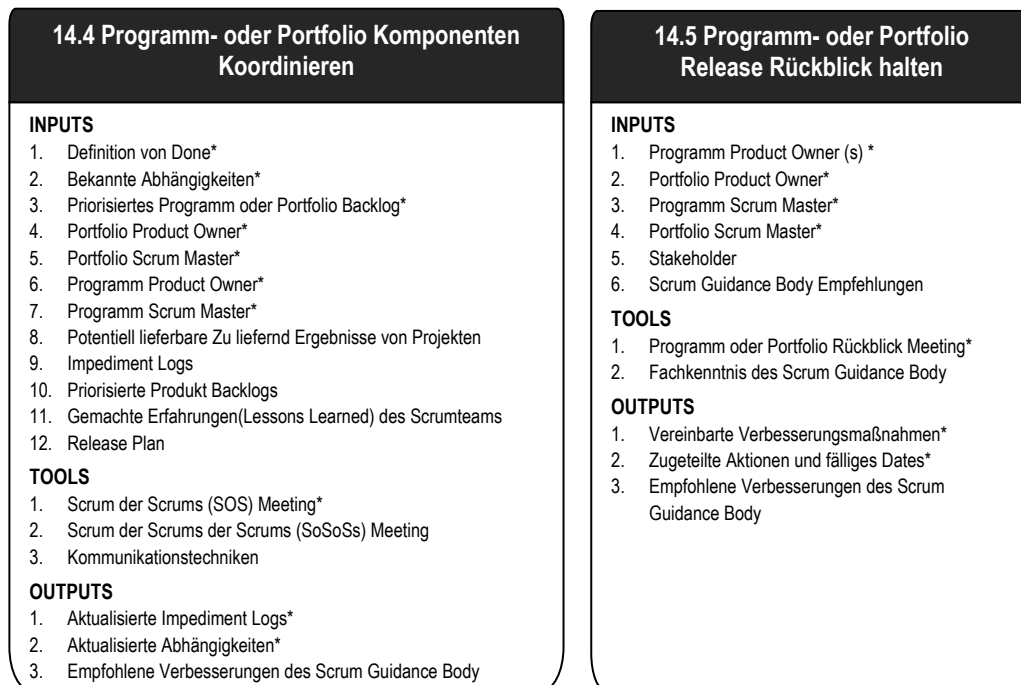
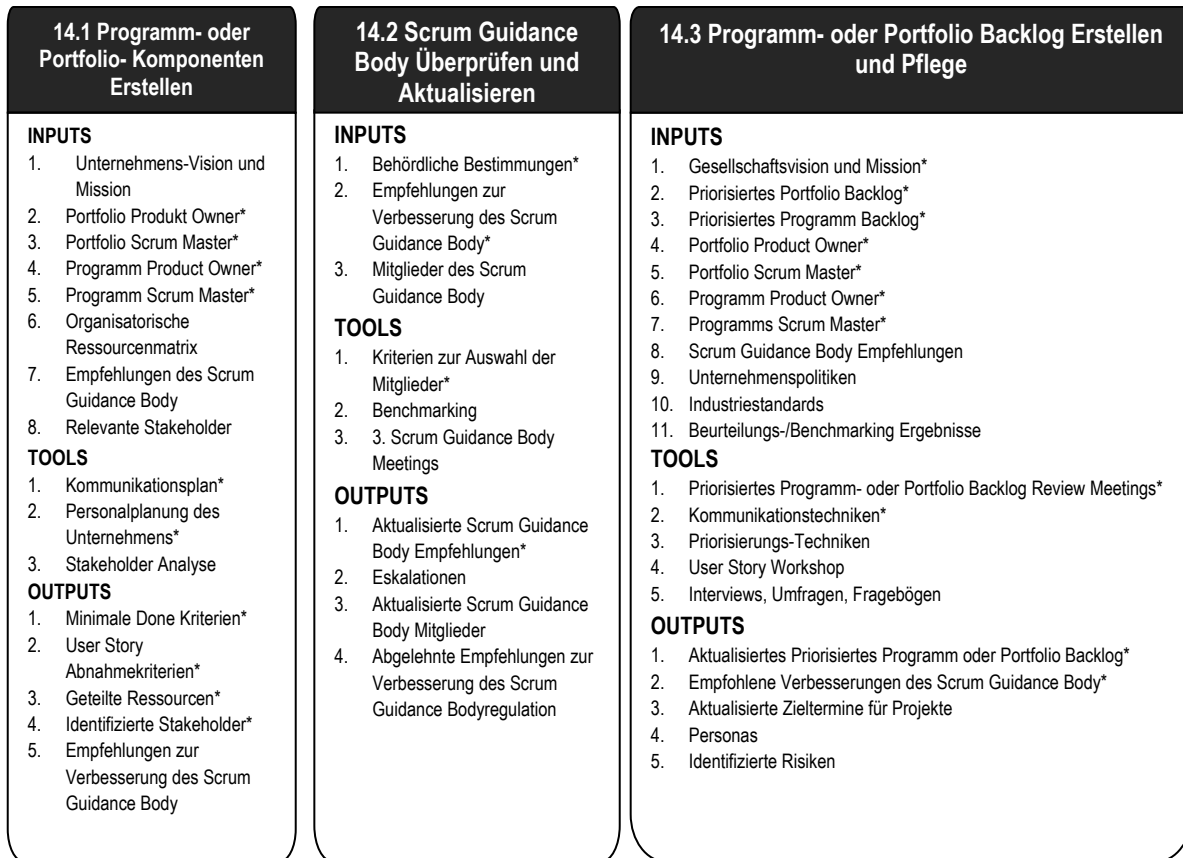


Abbildung 14-1 Scaling Scrum for the Enterprise Overview

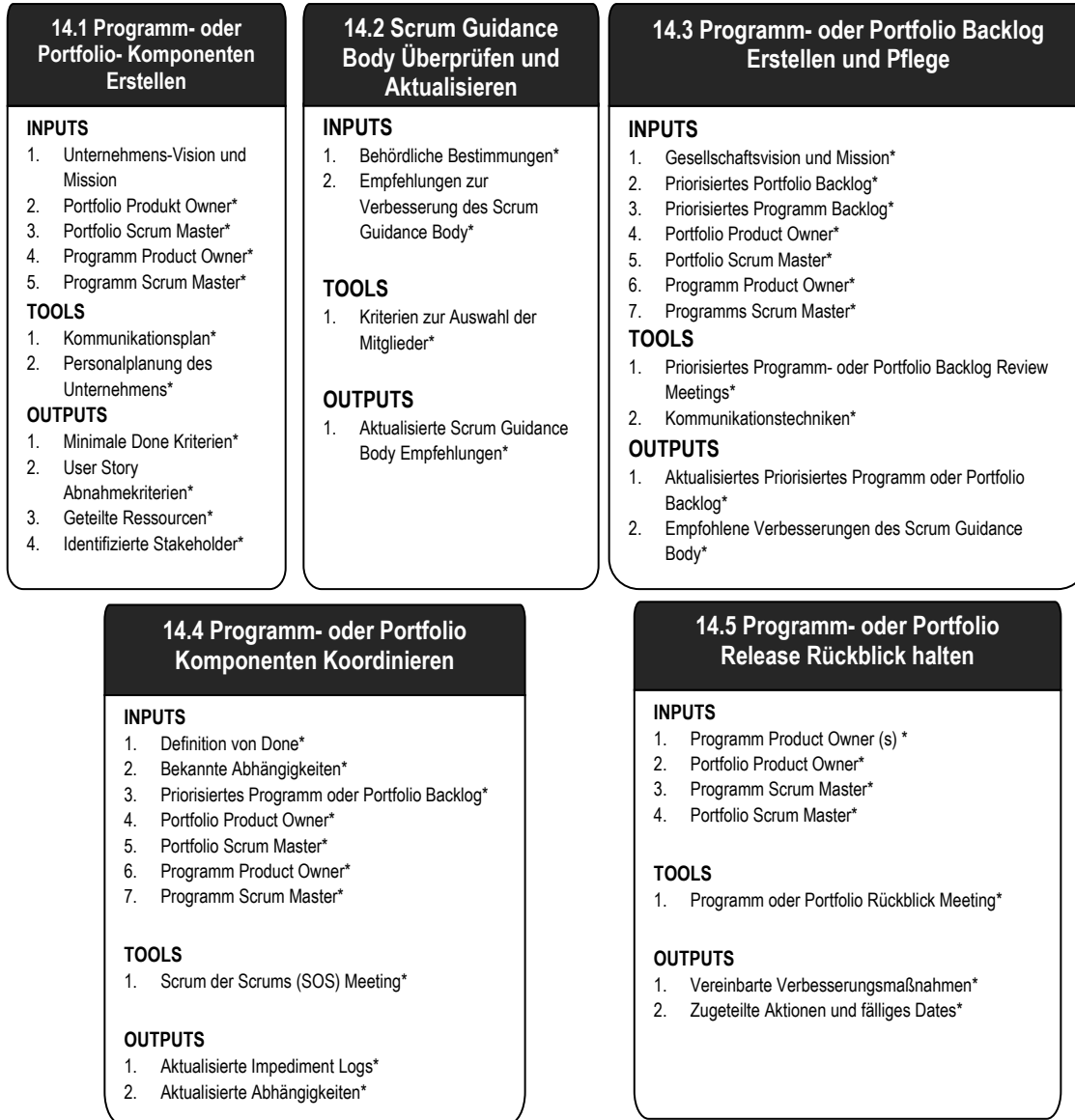


Abbildung 14-2 Scaling Scrum for the Enterprise Overview

14.1 Programm oder Portfolio Komponenten Erstellen

Abbildung 14-3 zeigt alle Inputs, Hilfsmittel und Outputs für den *Programm- oder Portfolio- Komponenten Erstellen* Prozess.

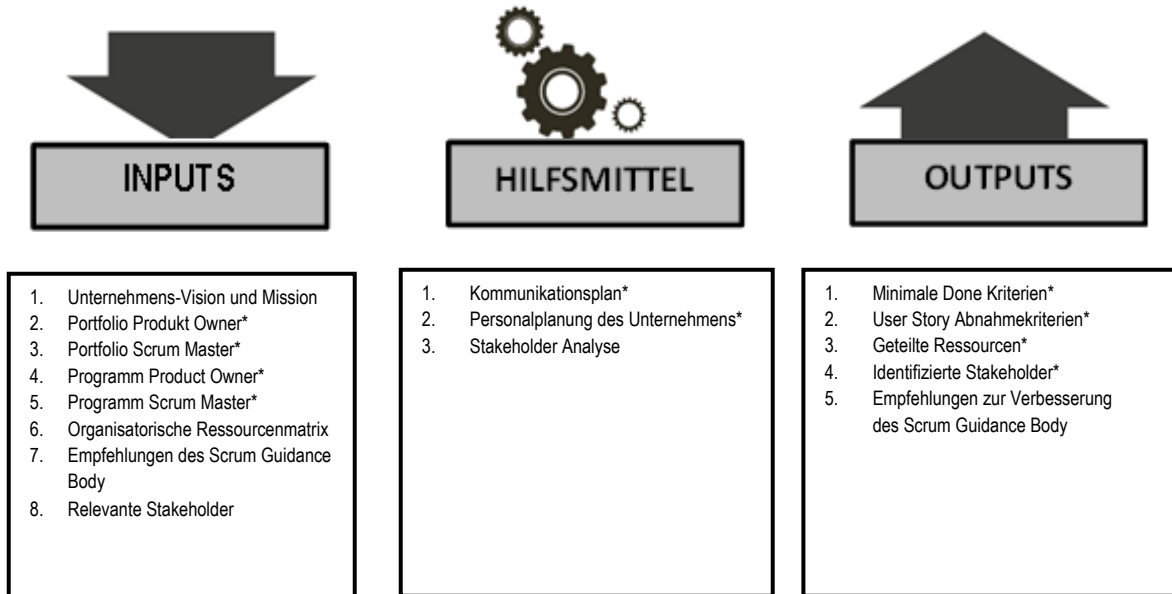


Abbildung 14 3: Programm- oder Portfolio- Komponenten Erstellen - Inputs, Hilfsmittel und Outputs

Hinweis - Sternchen (*) bezeichnen einen "verbindlichen" Input, ein "verbindliches" Hilfsmittel oder einen "verbindlichen" Output für den entsprechenden Prozess.

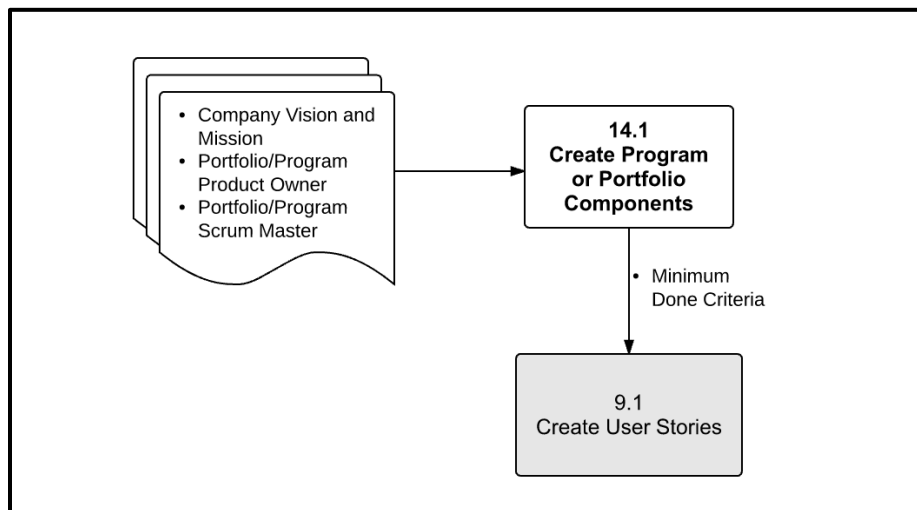


Abbildung 14-4: Programm- oder Portfolio- Komponenten Erstellen - Datenflussdiagramm

14.1.1 Inputs

14.1.1.1 Unternehmens- Vision und Mission

Unternehmens-Vision und Mission werden in Abschnitten 8.1.1.8 beziehungsweise 8.1.1.9 beschrieben.

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.

14.1.1.2 Portfolio Product Owner*

Beschrieben in Abschnitt 3.4.4.

14.1.1.3 Portfolio Scrum Master*

Beschrieben in Abschnitt 3.5.3.

14.1.1.4 Programm Product Owner*

Beschrieben in Abschnitt 3.4.3.

In einem Projekt ist der Programm Product Owner einer von mehreren Stakeholdern. Auf der Programmebene spielt der Programm Product Owner eine ähnliche Rolle wie der Product Owner in einem Projekt. Er/sie ist verantwortlich für die Erstellung und die Pflege von Programmkomponenten und treibt diese voran.

14.1.1.5 Programm Scrum Master*

Beschrieben in Abschnitt 3.5.2.

In einem Projekt ist der Programm Scrum Master einer von mehreren Stakeholdern. Auf Programmebene spielt der Programm Scrum Master eine ähnliche Rolle wie der Scrum Master in einem Projekt. Er/sie ist ein Vermittler, löst Probleme und entfernt Hindernisse auf der Programmebene.

14.1.1.6 Organisatorische Ressourcenmatrix

Beschrieben in Abschnitt 8.2.1.10.

14.1.1.7 Empfehlungen des Scrum Guidance Body

Beschrieben in Abschnitt 8.1.1.11 und 13.1.1.7.

Empfehlungen des Scrum Guidance Body sind auf Programm- und Portfolioebene besonders wichtig, da sie eine angemessene Anleitung für eine potentiell erhebliche Anzahl von Projekten liefern müssen.

14.1.1.8 Relevante Stakeholder

Während in diesem Prozess viele Stakeholder erst identifiziert werden, sind einige relevante Stakeholder mit Sicherheit schon vorher bekannt. Z. B. sind relevante Stakeholder auf der Portfolioebene die Mitglieder der Unternehmensführung oder staatliche Organisationen; relevante Stakeholder auf Programmebene sind z.B. der oder die Sponsor(en) des Programms oder zugehöriger Projekte sowie Senior Management.

Siehe zugehörige Abschnitte 1.4.3.1, 3.3.2 und 6.4.2.1.

14.1.2 Hilfsmittel

14.1.2.1 Kommunikationsplan*

Beschrieben in Abschnitt 12.1.2.2.

Der Kommunikationsplan sollte definieren, wie Informationen in der gesamten Organisation und im Portfolio und in den Programmen verbreitet wird. Er sollte auch definieren, wie und wann kommuniziert werden sollte, und welche Kommunikationsart bevorzugt ist. Das Portfolio liefert eine Anleitung für die Kommunikationspläne der Programme. In ähnlicher Weise liefert das Programm eine Anleitung für die Kommunikationspläne der Projekte.

14.1.2.2 Personalplanung des Unternehmens*

Die Personalplanung des Unternehmens liefert high-level Informationen darüber, wann bestimmtes Personal für verschiedene Projekte, Programme und Portfolios verfügbar sein wird. Sie liefert auch Informationen über Pläne für die Neueinstellung von Personal, das für zukünftige Projekte erforderlich sein wird.

14.1.2.3 Stakeholder Analyse

Die Standard Stakeholder Analyse wird dazu verwendet, die Stakeholder auf Programm- und Portfolioebene zu identifizieren. Weitere Details in Bezug auf Programm- oder Portfolio Stakeholder können als Personas im *Programm- oder Portfolio- Backlog Erstellen und Pflegen* Prozess identifiziert werden.

14.1.3 Outputs

14.1.3.1 Minimale Done Kriterien*

Beschrieben in Abschnitt 5.4.3.

14.1.3.2 User Story Abnahmekriterien*

Beschrieben in Abschnitt 9.1.3.2.

14.1.3.3 Geteilte Ressourcen*

Beschrieben in Abschnitt 13.1.3.4.

14.1.3.4 Identifizierte Stakeholder*

Beschrieben in Abschnitt 8.2.3.2.

Besonders relevante Stakeholder für das Portfolio oder Programm sind ein Input in diesen Prozess. Zusätzliche Stakeholder werden in diesem Prozess identifiziert.

14.1.3.5 Empfehlungen zur Verbesserung des Scrum Guidance Body

Beschrieben in Abschnitt 13.1.3.6.

Als Ergebnis des *Programm- oder Portfolio- Komponenten Erstellen* Prozesses könnten Vorschläge für potentielle Verbesserungen des Scrum Guidance Bodys (SGBs) geliefert werden. Diese empfohlenen Verbesserungen werden vom Scrum Guidance Body erörtert und entweder akzeptiert oder zurückgewiesen (siehe Abschnitt 14.2, Prozess *Scrum Guidance Body Überprüfen und Aktualisieren*). Wenn das SGB die Vorschläge akzeptiert, werden diese als Updates in die Scrum Guidance Body Dokumentation integriert.

14.2 Scrum Guidance Body Überprüfen und Aktualisieren

Abbildung 14-5 zeigt alle Inputs, Hilfsmittel und Outputs für den *Scrum Guidance Body Überprüfen und Aktualisieren* Prozess.

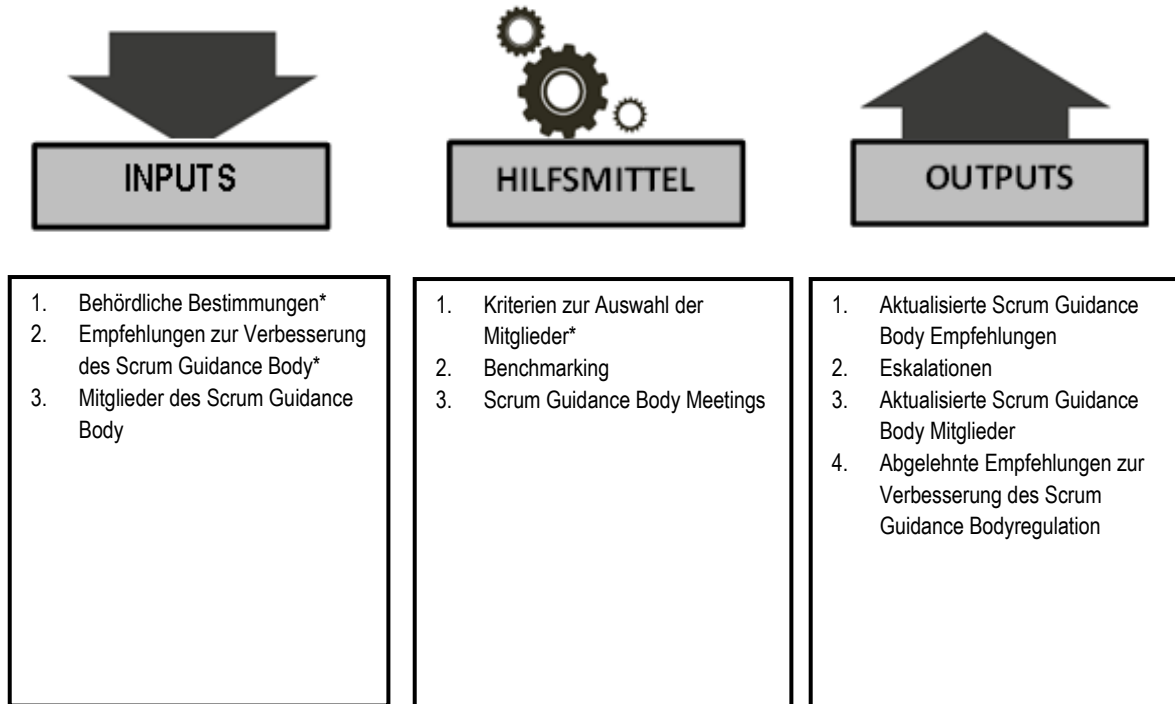


Abbildung 14-5: Scrum Guidance Body Überprüfen und Aktualisieren - Inputs, Hilfsmittel und Outputs

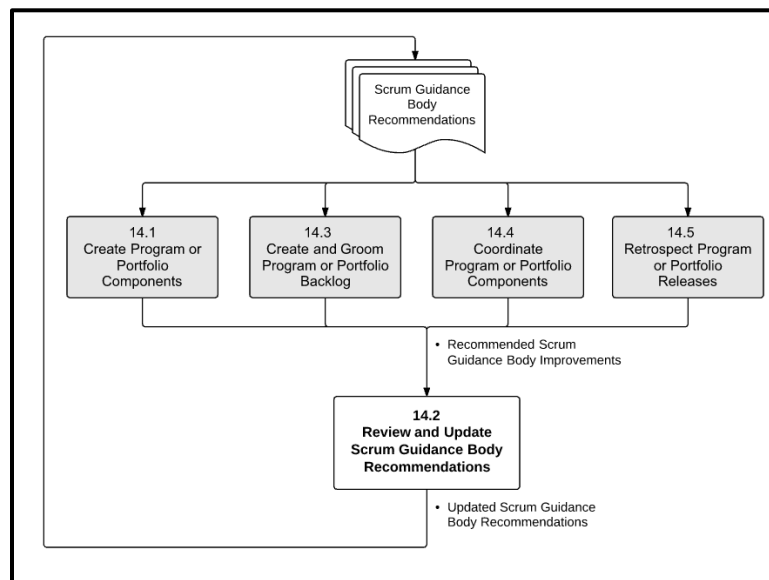


Abbildung 14-6: Scrum Guidance Body Überprüfen und Aktualisieren - Datenflussdiagramm

14.2.1 Inputs

14.2.1.1 Vorschriften*

Zu den Vorschriften gehören solche des Bundes, der Länder, der Städte und Gemeinden sowie des jeweiligen Industriezweigs, die das Programm oder Portfolio befolgen muss. User Storys, die erstellt wurden, um behördlichen Vorschriften innerhalb einer vorgegebenen Frist zu entsprechen, werden in das Portfolio oder Programm Produkt Backlog eingeschlossen.

Es ist durchaus möglich, dass Empfehlungen des Scrum Guidance Body aktualisiert werden müssen, um neue Vorschriften zu reflektieren.

14.2.1.2 Empfohlene Verbesserungen des Scrum Guidance Body*

Beschrieben in Abschnitt 13.1.3.6.

In Folge z. B. von Retrospektions-Meetings können Vorschläge gemacht werden, die Empfehlungen des Scrum Guidance Body zu überarbeiten oder zu verbessern. Wenn der Scrum Guidance Body diese Vorschläge akzeptiert, werden sie als Updates in die Empfehlungen des Scrum Guidance Body integriert.

14.2.1.3 Mitglieder des Scrum Guidance Body

Zu den Mitgliedern des Scrum Guidance Body- (SGB) können Scrumexperten, ausgewählte Scrum Master, Product Owner und Teammitglieder (auf allen Ebenen) gehören. Die Zahl der Mitglieder des SGB sollte jedoch begrenzt werden, um sicherzustellen, dass es relevant bleibt und keine zu restriktiven Empfehlungen publiziert.

14.2.2 Hilfsmittel

14.2.2.1 Kriterien zur Auswahl der Mitglieder*

Kriterien zur Auswahl der Mitglieder werden von den Stakeholdern festgelegt, um die Mitglieder des Scrum Guidance Body sowie ihre Rollen und Verantwortlichkeiten, ihre maximale Anzahl und ihre erforderliche Erfahrung und Sachkenntnis zu definieren.

Jedes Unternehmen kann und sollte seine eigenen Auswahlkriterien für Mitglieder des Scrum Guidance Body haben. Es wird jedoch dringend empfohlen, dass jedes Mitglied ein tiefgreifendes Verständnis von Scrum hat. Außerdem sollte, wie bereits erwähnt, die Zahl der Mitglieder begrenzt werden.

14.2.2.2 Benchmarking

Ein Unternehmen sollte seine eigenen Praktiken regelmäßig mit anderen Unternehmen vergleichen, um mit dem Wettbewerb Schritt zu halten. Benchmarking ist der Prozess, in dem die Geschäftsabläufe und Leistungs-Metriken einer Organisation mit jenen von führenden Gesellschaften in der eigenen oder anderen Industrien zu vergleichen.

14.2.2.3 Scrum Guidance Body Meetings

Der Scrum Guidance Body trifft sich regelmäßig, um den potentiellen Bedarf nach einer Aktualisierung der Empfehlungen des Scrum Guidance Body (z.B. von Retrospektions-Meetings empfohlene Verbesserungen, aktualisierte Bestimmungen, usw.) zu erörtern. Die Häufigkeit der Meetings wird vom Scrum Guidance Body basierend auf dem spezifischen Bedarf des Unternehmens festgelegt.

14.2.3 Outputs

14.2.3.1 Aktualisierte Empfehlungen des Scrum Guidance Body*

Beschrieben in 11.2.3.6.

In Folge der Überprüfung des Scrum Guidance Bodys können Änderungen notwendig sein und zu einer Aktualisierung der Empfehlungen des Scrum Guidance Body führen.

14.2.3.2 Eskalationen

Der Scrum Guidance Body mag feststellen, dass einige Richtlinien des Unternehmens die Teams daran hindern, maximalen Nutzen aus der Anwendung von Scrum zu ziehen. In einem solchen Fall sollte eine Eskalation mit dem Ziel, die Zustimmung zu einer Änderung der betreffenden Richtlinien zu erhalten, erfolgen.

14.3 Programm- oder Portfolio Backlog Erstellen und Pflegen

Abbildung 14-5 zeigt alle Inputs, Hilfsmittel und Outputs für den *Programm- oder Portfolio Backlog Erstellen und Pflegen* Prozess.

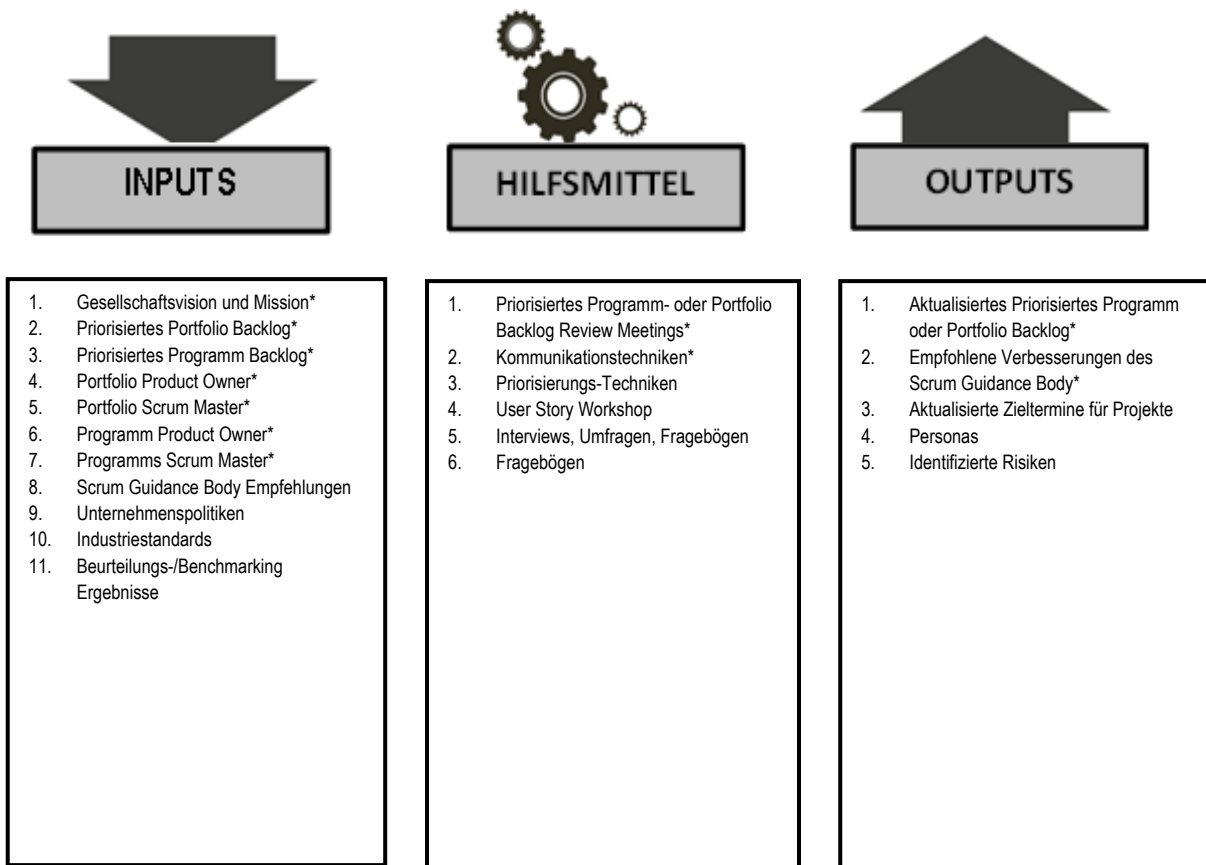


Abbildung 14-7: Programm- oder Portfolio Backlog Erstellen und Pflegen - Inputs, Hilfsmittel und Outputs

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

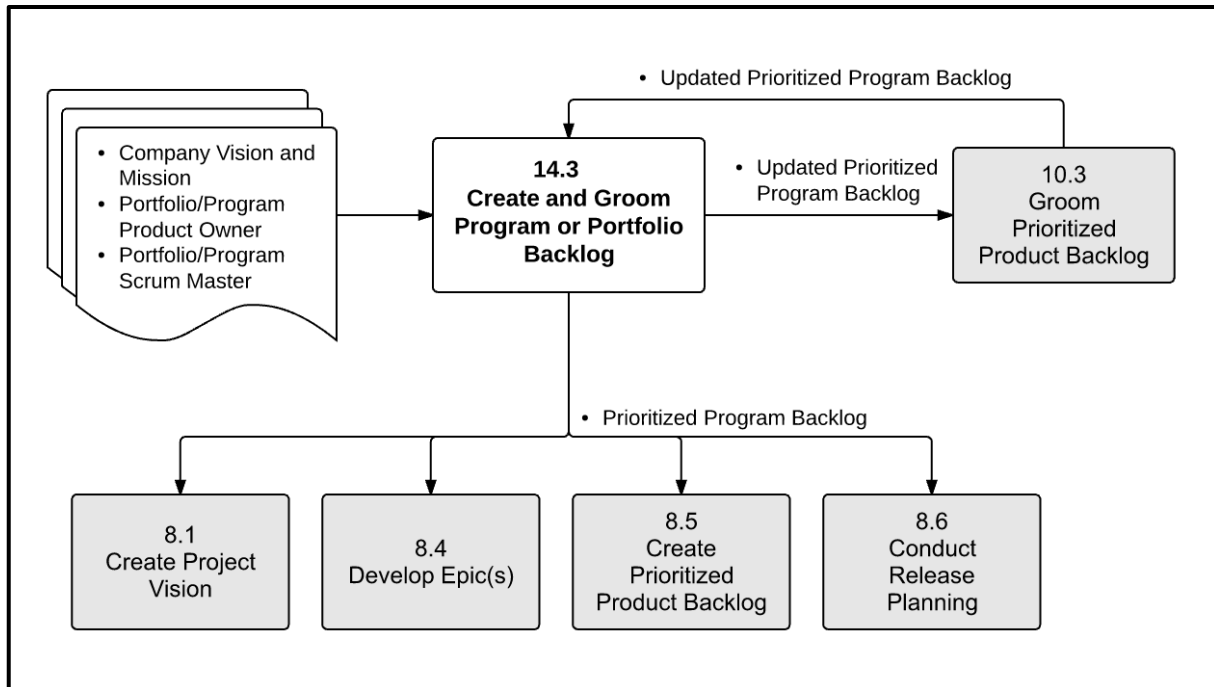


Abbildung 14-8: Programm- oder Portfolio Backlog Erstellen und Pflegen – Datenflussdiagramm

14.3.1 Inputs

14.3.1.1 Unternehmensvision und Mission*

Beschrieben in Abschnitten 8.1.1.8 und 8.1.1.9.

14.3.1.2 Priorisiertes Portfolio Backlog*

Das Priorisierte Portfolio Backlog spielt dieselbe Rolle auf Portfolioebene wie das Priorisierte Programm Backlog auf Programmebene. Die Einträge im Priorisierten Portfolio Backlog liefern Inputs für die verschiedenen Priorisierten Programm Backlogs und, über die Priorisierten Programm Backlogs, zu den Priorisierten Produkt Backlogs der einzelnen Projekte. Wie für Priorisierte Programmbacklogs in Abschnitt 8.1.1.5 beschrieben, werden Verfeinerungen von User Storys auf dieser Ebene, wenn überhaupt, nur minimal vorgenommen, da die Verfeinerung in den Projekten und ihren jeweiligen Priorisierten Produkt Backlogs stattfindet.

14.3.1.3 Priorisiertes Programm Backlog*

Beschrieben in Abschnitt 8.1.1.5.

Das Priorisierte Programm Backlog spielt dieselbe Rolle auf Programmebene wie das Priorisierte Produkt Backlog auf Projektebene. Es identifiziert die Erfordernisse für das Programm und ihre jeweiligen Prioritäten.

Es gibt allerdings einige Unterschiede:

Die Schaffung der jeweiligen zu liefernden Ergebnisse sowie ihre Abnahme werden in den jeweiligen Projekten des Programms durchgeführt. Die Done oder Abnahmekriterien für jeden Produkt Backlog Eintrag / jede User Story können auf der Programmebene definiert werden. Projekte müssen diese erfüllen, können aber ihre eigenen Kriterien hinzufügen.

Die Länge eines Sprints ist projektspezifisch und variiert im Allgemeinen von Projekt zu Projekt in einem Programm. Außerdem ist die Geschwindigkeit von verschiedenen Teams normalerweise nicht dieselbe. Deshalb ist es nicht notwendig, auf Programmebene User Storys sehr zu verfeinern. Die Verfeinerung von User Storys auf Programmebene geht nur weit genug, um sicherzustellen, dass die jeweilige User Story klar verstanden wird, und greifbare Abnahmekriterien für das Programm definiert werden können.

14.3.1.4 **Portfolio Product Owner***

Beschrieben in Sektion 3.4.4.

Der Portfolio Product Owner spielt eine ähnliche Rolle wie der Programm Product Owner in einem Programm. Er/sie ist verantwortlich und der Treiber für die Schaffung und die Pflege des Portfolio Produkt Backlogs.

14.3.1.5 **Portfolio Scrum Master***

Beschrieben in Sektion 3.5.3.

In einem Projekt ist der Programm Scrum Master einer von mehreren Stakeholdern. Der Portfolio Scrum Master spielt auf Portfolioebene eine ähnliche Rolle wie der Programm Scrum Master für ein Programm.

14.3.1.6 **Programm Product Owner***

Beschrieben in Sektion 3.4.3.

Der Programm Product Owner ist einer von mehreren Stakeholdern in einem Projekt. Auf Programmebene spielt der Programm Product Owner eine ähnliche Rolle wie der Product Owner in einem Projekt. Er/sie ist verantwortlich und der Treiber für die Schaffung und die Pflege des Priorisierten Programm Produkt Backlogs.

14.3.1.7 **Programm Scrum Master***

Beschrieben in Sektion 3.5.2.

Der Programm Scrum Master ist einer von mehreren Stakeholdern in einem Projekt. Auf Programmebene spielt der Programm Scrum Master eine ähnliche Rolle wie der Scrum Master in einem Projekt. Er/sie ist ein Vermittler, löst Probleme und beseitigt Hindernisse auf Programmebene.

14.3.1.8 **Empfehlungen des Scrum Guidance Body**

Beschrieben in Abschnitt 8.1.1.11 und 10.3.1.11.

Bei der Schaffung und Pflege eines Priorisierten Programm- oder Portfolio Backlogs liefern die Empfehlungen des Scrum Guidance Body beste Praktiken, die auf Programm- oder Portfolioebene in Betracht gezogen werden sollten.

14.3.1.9 **Unternehmensrichtlinien**

Unternehmensrichtlinien bestehen aus den Prinzipien, Regeln und Anleitungen, die von einer Organisation formuliert oder übernommen wurden. Die Änderung von Unternehmensrichtlinien würde geschaffene User Storys beeinflussen, da bei ihrer Erstellung bestehende Richtlinien angewendet wurden.

14.3.1.10 **Industriestandards**

Neue Industriestandards oder Änderungen an vorhandenen Standards müssen implementiert werden, um die Lebensfähigkeit eines Produkts oder Dienstes zu erhalten. Deshalb müssen entsprechende User Storys in das Priorisierte Programm- und/oder Portfolio- Backlog eingebracht und entsprechend priorisiert werden.

Manchmal müssen Empfehlungen des Scrum Guidance Body angepasst werden, um neue oder geänderte Standards zu reflektieren.

14.3.1.11 **Ergebnisse von Beurteilungen/Benchmarking**

In erster Linie machen die Ergebnisse von Beurteilungen/Benchmarking eine Aktualisierung von in den Empfehlungen des Scrum Guidance Body enthaltenen Besten Praktiken erforderlich. Sie können auch helfen, einen Minimalstandard beim Schaffen eines Produkts oder eines Dienstes festzulegen, und auch zu geänderten Done Kriterien führen. Manchmal können sie auch Programm- oder Portfolio Product Ownern den Anstoß geben, neue User Storys zur Implementierung der Besten Praktiken zu entwickeln.

14.3.2 Hilfsmittel

14.3.2.1 Priorisiertes Programm- oder Portfolio Backlog Review Meetings*

Beteiligung an Programm- oder Portfolio Backlog Review Meetings unterscheidet sich ziemlich von der Beteiligung an Review Meetings auf Projektebene. Scrumteams nehmen an den Pflege Sitzungen auf Projektebene teil. Auf Programm- oder Portfolioebene ist jedes Projekt im Programm oder jedes Programm im Portfolio vertreten. Um das Meeting zu vereinfachen ist es normalerweise empfehlenswert, nur einen Vertreter jedes Projekts oder jedes Programms bei diesen Review Meetings zu haben.

Siehe verwandte Abschnitte 10.3.2.1 und 6.5.1.2.

14.3.2.2 Kommunikationstechniken*

Beschrieben in Abschnitt 10.3.2.2.

14.3.2.3 Priorisierungstechniken

Beschrieben in Abschnitt 8.5.2.1.

Auf Programm- oder Portfolioebene gibt es normalerweise eine geringere Anzahl von Anforderungen/User Storys als auf Projektebene. Der Prozentsatz von User Storys, die einen sehr greifbaren Wert oder Geschäftsbedarf oder Benutzerwirkung haben, ist normalerweise ebenfalls viel kleiner als auf Projektebene. Das bedeutet, dass die Auswahl an Techniken, die auf Programm- oder Portfolioebene nützlich sind, ebenfalls kleiner ist.

Zum Beispiel ist die Kano Analyse nur von eingeschränktem Nutzen, weil es keine Begeisterungs- oder Ablehnungs-Features gibt. Ohne eine große Anzahl von Stakeholdern, insbesondere Anwendern, hat auch die 100 Punkt Methode nur beschränkten Wert. Die MoSCoW Technik hat ebenfalls Beschränkungen, weil es auf Programm- und Portfolioebene keine "could have" oder "won't have" Features gibt.

Paarweiser Vergleich funktioniert auf Programm- und Portfolioebene hingegen gut.

14.3.2.4 User Story Workshop

Beschrieben in Abschnitt 8.4.2.2.

Verglichen mit Projekten produzieren User Story Workshops für Programme und Portfolios nur high-level User Storys als Outputs, und die Anzahl von User Storys wird auch niedriger sein. Jedoch liefern diese Meetings immer noch Mehrwert, weil es Beteiligung von Vertretern von verschiedenen Projekten in einem

Programm oder von verschiedenen Programmen in einem Portfolio gibt. Damit wird sichergestellt, dass Anforderungen wohldefiniert und verstanden sind.

14.3.2.5 Interviews, Studien

Beschrieben in Abschnitten 8.4.2.4.

14.3.2.6 Fragebögen

Beschrieben in Abschnitt 8.4.2.5.

14.3.3 Outputs

14.3.3.1 Aktualisiertes Priorisiertes Programm oder Portfolio Backlog*

Das Priorisierte Programm- oder Portfolio Backlog kann mit neuen User Storys, neuen Change Requests, neuen identifizierten Risiken, aktualisierten User Storys oder Re-Priorisierung von vorhandenen User Storys aktualisiert werden.

Siehe auch Abschnitt 10.3.3.1.

14.3.3.2 Empfohlene Verbesserungen des Scrum Guidance Body*

Beschrieben in Abschnitt 13.1.3.6.

Als Ergebnis des Prozesses Programm- oder Portfolio Backlog Erstellen und Pflegen könnten Vorschläge oder Rückmeldungen für potentielle Verbesserungen des Scrum Guidance Bodys gemacht werden. Diese empfohlenen Verbesserungen werden vom Scrum Guidance Body erörtert und entweder akzeptiert oder zurückgewiesen (siehe Abschnitt 14.2, Scrum Guidance Body Reviewen und Aktualisieren Prozess). Wenn der Scrum Guidance Body diese Vorschläge akzeptiert, werden sie als Aktualisierungen in die Dokumentation des Scrum Guidance Body integriert.

14.3.3.3 Aktualisierte Implementierungsfristen für Projekte

Implementierungsfristen für Projekte werden gegebenenfalls aktualisiert, um die Auswirkungen von neuen oder geänderten User Storys zu reflektieren, die Anforderungen modifizieren oder neue Anforderungen einführen.

14.3.3.4 **Personas**

Beschrieben in Abschnitt 8.4.3.2.

14.3.3.5 **Identifizierte Risiken**

Beschrieben in Abschnitt 8.4.3.4.

14.4 Programm- oder Portfolio Komponenten Koordinieren

Abbildung 14-6 zeigt alle Inputs, Hilfsmittel und Outputs für den Programm- oder Portfolio Komponenten Koordinieren Prozess.

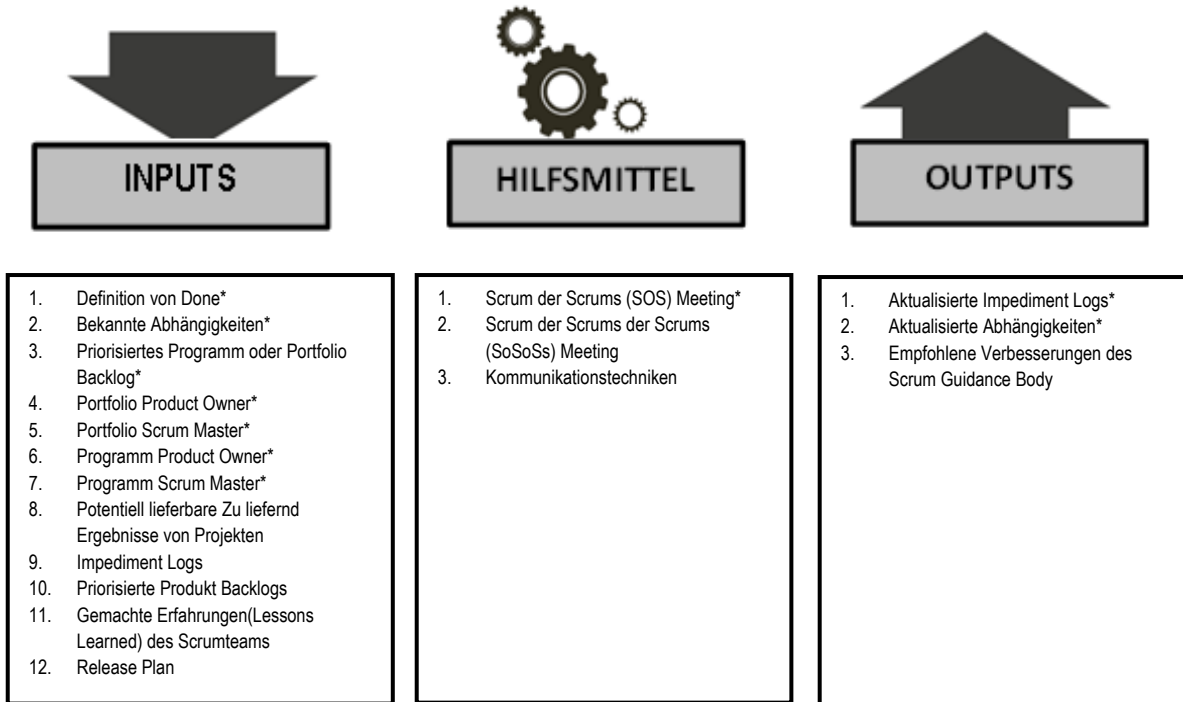


Abbildung 14-9: Programm- oder Portfolio Komponenten Koordinieren Inputs, Hilfsmittel und Outputs

Hinweis Sternchen (*) bezeichnen einen "verbindlichen" Input, ein "verbindliches" Hilfsmittel oder einen "verbindlichen" Output für den entsprechenden Prozess.

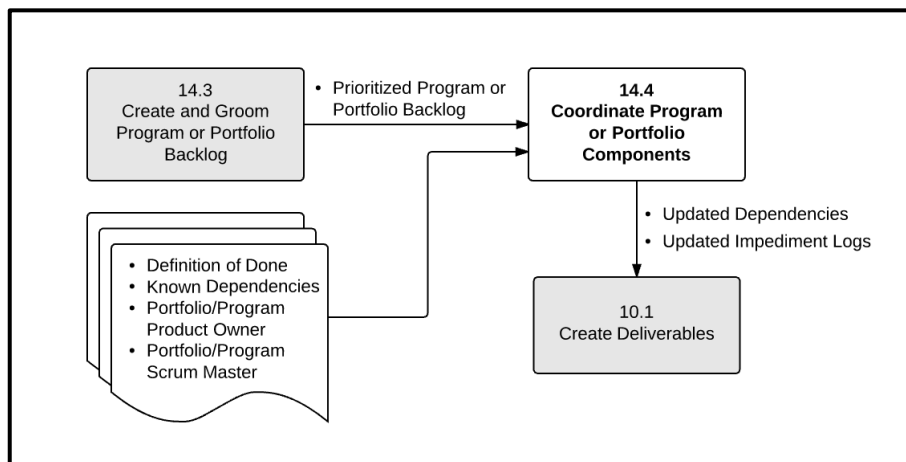


Figure 14-1: Coordinate Program or Portfolio Components—Data Flow Diagram

14.4.1 Inputs

14.4.1.1 Definition von Done*

Beschrieben in Abschnitt 5.4.3.

Die Definition von Done auf Programm oder Portfolio Ebene kann als Minimale Done Kriterien für Projekte des ganzen Unternehmens benutzt werden.

14.4.1.2 Bekannte Abhängigkeiten*

Im Falle von zusammengehörigen Projekten und/oder Produkten innerhalb des Unternehmens kann es bekannte Abhängigkeiten zwischen ihnen geben. Es sollte deshalb eine projektübergreifende Koordinierung stattfinden, um diese Abhängigkeiten zu handhaben. Hier einige Beispiele für solche Abhängigkeiten:

- Dasselbe Release Datum für zusammengehörige Projekte
- Abhängigkeiten zwischen Releases
- Abhängigkeiten von zusammengehörigen Features

Siehe zugehörige Abschnitte 9.4.2.3 und 13.1.2.4.

14.4.1.3 Priorisiertes Programm oder Portfolio Backlog*

Beschrieben in Abschnitten 14.3.1.2 und 14.3.1.3.

14.4.1.4 Portfolio Product Owner*

Beschrieben in Abschnitten 3.4.4 und 14.3.1.4.

14.4.1.5 Portfolio Scrum Master*

Beschrieben in Abschnitten 3.5.3 und 14.3.1.5.

14.4.1.6 Programm Product Owner*

Beschrieben in Abschnitten 3.5.2 und 14.3.1.6.

14.4.1.7 Programm Scrum Master*

Beschrieben in Abschnitten 3.5.2 und 14.3.1.7.

14.4.1.8 Potentiell lieferbare Zu liefernde Ergebnisse von Projekten

Potentiell lieferbare Zu liefernde Ergebnisse von Projekten sind wertvolle Inputs für die Koordinierung auf Programm- oder Portfolioebene. Am Ende von Sprints in Projekten sind Produktinkremente oder Zu liefernde Ergebnisse komplett. Die in diese Inkremente einbezogenen User Storys erfüllen sowohl die Done Kriterien als auch ihre jeweiligen Abnahmekriterien.

Mit jedem abgeschlossenen Sprint in jedem der zugehörigen Projekte wird klarer, wie die jeweiligen Projekte voranschreiten. Dieses Wissen erlaubt nicht nur stetig aktualisierte Voraussagen, ob alle Projekte die erforderlichen Fristen von bestimmten Anforderungen einhalten werden, sondern es liefert auch wichtige Informationen, um die Abhängigkeiten zwischen Projekten zu handhaben.

14.4.1.9 Impediment Logs

Hindernisse, denen sich einzelne Projekte gegenüber sehen, können für andere Projekte relevant sein. Deshalb kann es sein, dass Impediment Logs unter Projekten und/oder Programmen geteilt werden müssen.

Beschrieben in Abschnitt 10.1.1.4. Siehe zugehörigen Abschnitt 13.2.1.2 für das Teilen von Impediment Logs zwischen Teams in einem großen Projekt.

14.4.1.10 Priorisierte Produkt Backlogs

Beschrieben in Abschnitt 8.5.3.1.

14.4.1.11 Gemachte Erfahrungen (Leasons Learned) des Scrum Teams

Beschrieben in Abschnitt 11.3.3.5.

14.4.1.12 Release Pläne

Beschrieben in Abschnitt 8.6.3.1.

Diese Pläne sind, obwohl sie vorläufig und potentiellen Änderungen unterworfen sind, sehr wichtig, um abzuschätzen, ob die jeweiligen Projekte wahrscheinlich erforderliche Fristen einhalten. Sie sind entscheidend in Bezug auf Abhängigkeiten.

14.4.2 Hilfsmittel

14.4.2.1 Scrum der Scrums (SOS) Meeting*

Beschrieben in Abschnitt 13.2.2.1.

Dieses Meeting wird sehr ähnlich zu seiner Verwendung bei großen Projekten verwendet. Auf Programm Ebene kommen Vertreter aller Projekte im Programm zu regelmäßigen Meetings zusammen (Scrum der Scrums oder SoS).

14.4.2.2 Scrum der Scrum der Scrums (SoSoS) Meeting

Auf Portfolio Ebene kommen Vertreter aller Programme und eigenständigen Projekte im Portfolio in regelmäßigen Abständen zusammen. Man könnte auch sagen, dass sich Vertreter von jedem Scrum der Scrums treffen. Deshalb wird diese zusätzliche Ebene von Meetings auch Scrum der Scrums der Scrums genannt (SoSoS.) Abbildung 3-4 veranschaulicht das Konzept eines Scrum der Scrums Meeting.

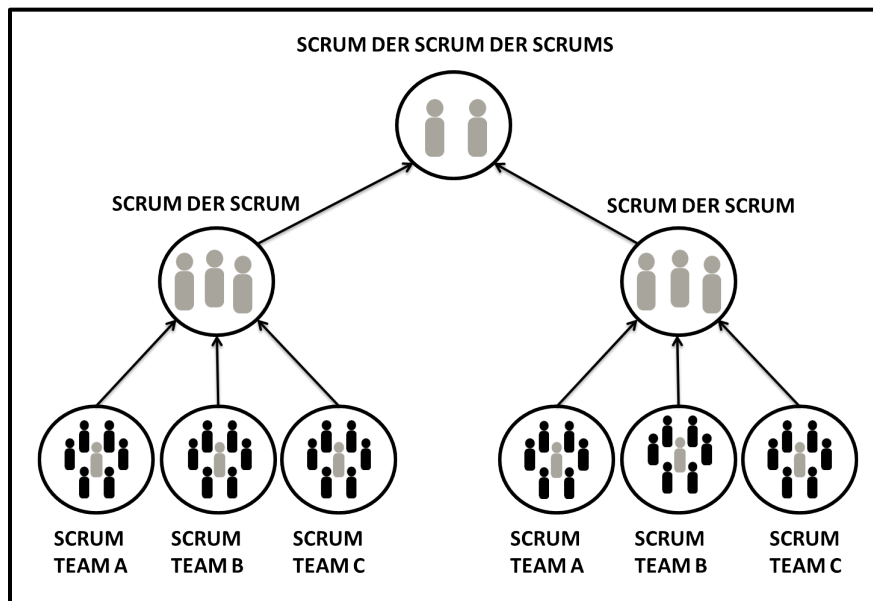


Abbildung 14-11: Scrum der Scrum der Scrums (SoS) Meeting

In diesem Beispiel werden sechs Projekte gleichzeitig durchgeführt. Drei der Projekte stehen in direktem Bezug zueinander: Projekte A, B und C. Die anderen drei direkt miteinander in Verbindung stehenden

Projekte sind D, E und F. Ein Scrum der Scrums Meeting wird abgehalten, um die gegenseitigen Abhängigkeiten der direkt miteinander verbundenen Projekte jeder Gruppe zu koordinieren. Danach findet ein Scrum der Scrum der Scrums Meeting statt, um die Tätigkeiten aller Projekte zu koordinieren und zu managen

14.4.2.3 Kommunikationstechniken

Beschrieben in Abschnitt 10.3.2.2.

Manche der Kommunikationstechniken, die auf Programm- oder Portfolioebene verwendet werden können, sind Message Boards und Instant Messaging.

14.4.3 Outputs

14.4.3.1 Aktualisierte Impediment Logs*

Als Ergebnis des Scrum der Scrums (SOS) oder des Scrum der Scrum der Scrums (SoSoS) Meetings kann es notwendig sein, die Impediment Logs zu aktualisieren.

Siehe zugehörigen Abschnitt 10.1.1.4.

14.4.3.2 Aktualisierte Abhängigkeiten*

Als Ergebnis der Koordinierung von Programm- oder Portfolio Komponenten kann es notwendig sein, den bekannten Abhängigkeiten neue Abhängigkeiten hinzuzufügen oder vorhandene Abhängigkeiten zu aktualisieren.

Siehe zugehörige Abschnitte 9.4.3.3 und 13.1.3.9.

14.4.3.3 Empfohlene Verbesserungen des Scrum Guidance Body*

Beschrieben in Abschnitt 13.1.3.6.

Als Ergebnis des Prozesses Programm- oder Portfolio Komponenten Koordinieren könnten Vorschläge oder Rückmeldungen für potentielle Verbesserungen des Scrum Guidance Bodys gemacht werden. Diese empfohlenen Verbesserungen werden vom Scrum Guidance Body erörtert und entweder akzeptiert oder zurückgewiesen (siehe Abschnitt 14.2, Scrum Guidance Body Reviewen und Aktualisieren Prozess). Wenn

der Scrum Guidance Body diese Vorschläge akzeptiert, werden sie als Aktualisierungen in die Dokumentation des Scrum Guidance Body integriert.

14.5 Programm- oder Portfolio- Release Rückblick Halten

Abbildung 14-12 zeigt alle Inputs, Hilfsmittel und Outputs für den Programm- oder Portfolio Release Rückblick halten Prozess.

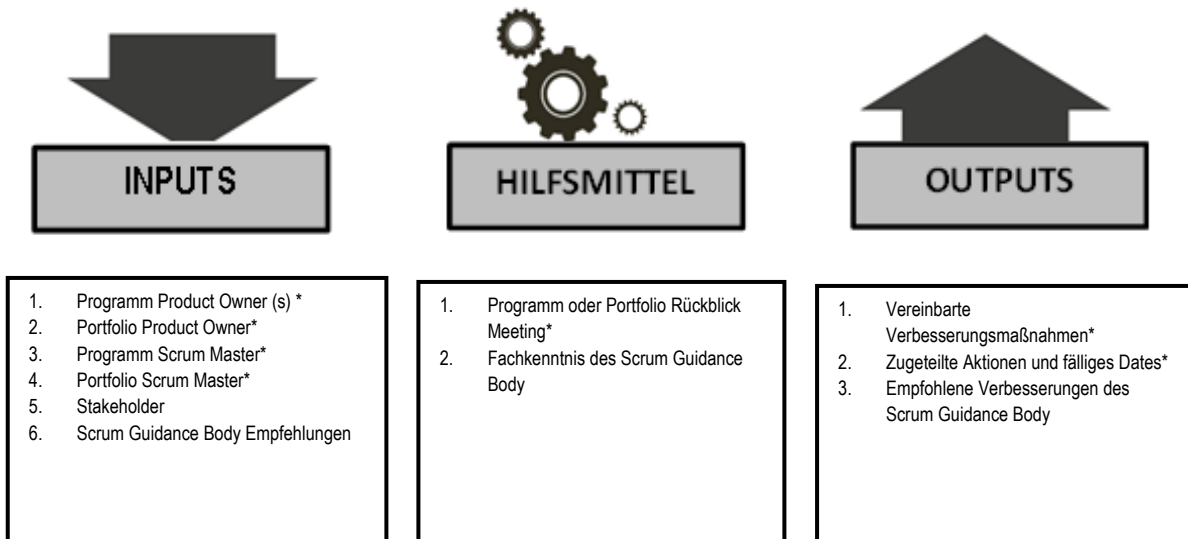


Abbildung 14-12: Programm- oder Portfolio- Release Rückblick Halten - Inputs, Hilfsmittel und Outputs

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

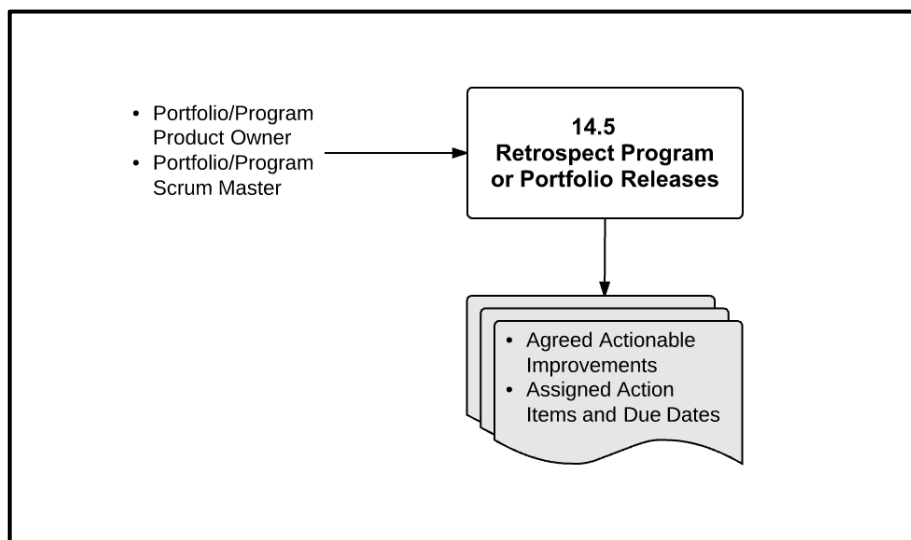


Abbildung 14-13: Programm- oder Portfolio-Release Rückblick Halten —Datenflussdiagramm

14.5.1 Inputs

14.5.1.1 Portfolio Product Owner*

Beschrieben in Abschnitten 3.4.4 und 14.3.1.4.

14.5.1.2 Portfolio Scrum Master*

Beschrieben in Abschnitten 3.5.3 und 14.3.1.5.

14.5.1.3 Programm Product Owner*

Beschrieben in Abschnitten 3.5.2 und 14.3.1.6.

14.5.1.4 Programm Scrum Master*

Beschrieben in Abschnitten 3.5.2 und 14.3.1.7.

14.5.1.5 Stakeholder

Beschrieben in Abschnitt 3.3.2.

14.5.1.6 Empfehlungen des Scrum Guidance Body

Beschrieben in Abschnitt 8.1.1.11 und 12.2.1.5.

Während einer Retrospektive von Programm- oder Portfolio- Releases liefern die Empfehlungen des Scrum Guidance Body beste Praktiken, die sich auf wichtige Themen wie Verwaltungsverfahren, Audits, Bewertungen und Projekt-Übergangs-Kriterien beziehen. Das ist der Anwendung von vom Scrum Guidance Body gelieferten Empfehlungen in den Retrospektiven auf Projektebene ähnlich (beschrieben in Abschnitt 12.2.1.5).

14.5.2 Hilfsmittel

14.5.2.1 Programm oder Portfolio Retrospektions Meeting*

Das Programm oder Portfolio Retrospektions Meeting ist Projekt Retrospektions Meeting, beschrieben in 12.2.2.1, ähnlich, wird aber auf dem Programm- oder Portfolio Level. Der wesentliche Unterschied ist, dass Programm und Portfolio Retrospektions Meetings viel seltener stattfinden als Projekt Retrospektions Meetings.

14.5.2.2 Fachkenntnis des Scrum Guidance Body

Beschrieben in Abschnitt 8.4.2.7.

14.5.3 Outputs

14.5.3.1 Vereinbarte Verbesserungsmaßnahmen*

Beschrieben in Abschnitt 11.3.3.1.

14.5.3.2 Zugeordnete Aktionspunkte und Fertigstellungstermine*

Beschrieben in Abschnitt 11.3.3.2

14.5.3.3 Empfohlene Verbesserungen des Scrum Guidance Body

Beschrieben in Abschnitt 13.1.3.6.

Als Ergebnis des Prozesses Programm- oder Portfolio Release Rückblick Halten könnten Vorschläge oder Rückmeldungen für potentielle Verbesserungen des Scrum Guidance Bodys gemacht werden. Diese empfohlenen Verbesserungen werden vom Scrum Guidance Body erörtert und entweder akzeptiert oder zurückgewiesen (siehe Abschnitt 14,2, Scrum Guidance Body Reviewen und Aktualisieren Prozess). Wenn der Scrum Guidance Body diese Vorschläge akzeptiert, werden sie als Aktualisierungen in die Dokumentation des Scrum Guidance Body integriert.

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 angewendeten 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 populären 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:

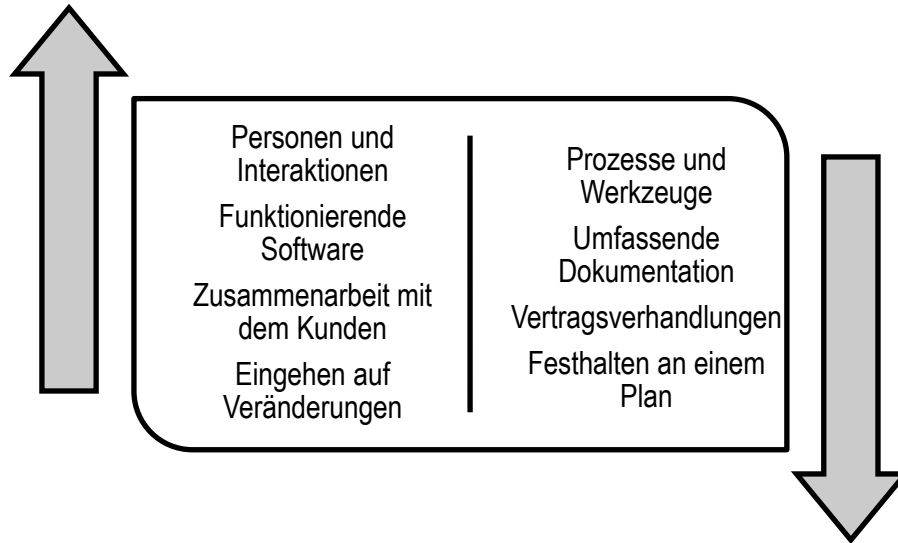
1. Wettbewerbsintensive Märkte
2. Niedrigere Kosten für Experimente
3. Ständiger Fortschritt der Technologie
4. Kürzere Produktlebenszyklen
5. Wissensarbeiter ersetzen Facharbeiter

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 Stakeholder voneinander abhängig und miteinander verbunden sind und 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

J. Drew Nations

Buddy Peacock

Ruth Kim

Deepak Ramaswamy

B.3 Rezensenten und Editoren Team

James Pruitt

Elizabeth Lynne Warren

Syed Ashraf

Melvin Wofford Jr.

Srinivas Reddy Kandi

Girish Kulkarni

Abdelnaser Dwaikat

Anu Ravi

Morris Feigel

Ian Glenister

Jo Pereira
Saurabh Gupta
Nikhil Bhargava
Simon Robertson
Dusan Kamenov
Prof. Dr. Akram Hassan
Meena Elangovan
Olumide Idowu
Vinod Kumar
Michael Rauch
Joe Schofield

B4 Früher Ausgabe

Co-Autoren und Fachexperten

R-A Alves
Winfried Hackmann
Quincy D. Jordan
Gaynell Malone
J. Drew Nations
Buddy Peacock
Karen Lyncook
Jaimie M. Rush
Elizabeth Lynne Warren
Ruth Kim
Mehul Doshi
Gaurav Garg
Ajey Grandhem
Sayan Guha
Vinay Jagannath
Deepak Ramaswamy
Ahmed Touseefullah Siddiqui

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.

Spears, L. C. (2010) "Character and Servant Leadership: Ten Characteristics of Effective, Caring Leaders." *The Journal of Virtues & Leadership*, 1 (1): 25–30.

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 Stakeholder 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 Stakeholder 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 Stakeholders 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 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, Stakeholder 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 Stakeholder. 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 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 Stakeholder 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 verbundenen 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 erstellen

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 Stakeholder 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 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 erfolglose 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/ihres 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 pflegen

"Priorisiertes Produkt Backlog pflegen" ist ein Prozess, in dem das Priorisierte Produkt Backlog ständig aktualisiert und gepflegt wird.

Prozess zur Identifizierung von Scrum Master und Stakeholder(n)

Prozess, in dem Scrum Master und Stakeholder 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 einen 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 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 Produktanpassungen 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 Stakeholder 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 Stakeholder 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

Laissez Faire Leader

Ein Führungsstil, der das Team weitestgehend unbeaufsichtigt und ohne Eingriff des Leaders in den Arbeitsalltag belässt. Dieser Stil führt häufig zu einem anarchischen Zustand.

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, Zielanwender 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 Stakeholder, 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 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 erstellen

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 Stakeholder (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 pflegen" 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-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 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 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 Stakeholder 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.

Projekt Rückblick halten

In diesem Prozess, der das Projekt abschließt, kommen organisatorische Stakeholder 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 Stakeholder 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 Stakeholder, wieviel Risiko für den/die Stakeholder akzeptabel ist. Risikoverhalten ist ein entscheidender Faktor, wenn Stakeholder eine Entscheidung hinsichtlich Maßnahmen zur Minderung potenzieller negativer Risiken treffen.

Risikoscheu

Risikoscheu ist eine der Kategorien der Nutzenfunktion. Sie bezieht sich auf einen Stakeholder, 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 Stakeholder 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 Stakeholder 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 Stakeholder bezieht, der weder risikoscheu noch risikosuchend ist. Getroffene Entscheidungen sind nicht von der Unwägbarkeit des Ergebnisses beeinflusst. Wenn zwei mögliche Szenarien den gleichen Nutzen bringen, wird sich der risikoneutrale Stakeholder 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 Instrustriezweige und Projekttypen zur Verfügung.

Risikosuchend

Risikosuchend ist einer der Kategorien der Nutzenfunktion, die sich auf einen Stakeholder 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 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 Stakeholder 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 "unerledigt" 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.

Dienender Leader

Dienende 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.

Zu liefernde Ergebnisse ausliefern

In diesem Prozess werden Akzeptierte zu liefernde Ergebnisse ausgeliefert oder an den zuständigen Stakeholder 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 Stakeholder 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 Erstellen" 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

Stakeholder

Stakeholder 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 Stakeholder und Anwender ins Auge. Die Stakeholder 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 nieder prior 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 Stakeholder, 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 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 Stakeholder einzubeziehen.

User Story Schreibfachkenntnis

Basierend auf seiner oder ihrer Interaktion mit den 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 Stakeholder. Es definiert den Level der Risikobereitschaft der Stakeholder.

Wertstromanalyse

Wertstromanalyse nutzt Flussdiagramme, um den zur Vervollständigung eines Prozesses erforderlichen Informationsfluss zu illustrieren. Sie kann dazu verwendet werden, einen Prozess durch Identifizierung der nicht-wertschöpfenden Elemente zu straffen.

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 dritte Auflage des *SBOK™* ergänzt die vorherige Ausgabe in den Kapiteln dreizehn und vierzehn mit erweitertem Inhalt zum Thema der Skalierung von Scrum, sowohl für große Projekte als auch für das Unternehmen.

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.

