

Gerd Bauer – Dierk Söllner – Hans-Jörg Steffe



# Certified Scrum Practitioner

Syllabus Version 04.00 deutschsprachige Version 01.00



Enthält die Syllabi für

- Scrum Master
- Product Owner

itedas.org  
Reginbaldstr.12  
81247 München  
www.itedas.org  
ServiceDesk@itedas.org



# Inhaltsverzeichnis

<b>Einführung</b> .....	<b>3</b>
<b>Zielgruppe</b> .....	<b>3</b>
<b>Versionshistorie</b> .....	<b>4</b>
<b>Certified Scrum Practitioner (CSP) Curriculum Überblick</b> .....	<b>5</b>
Product Owner Seminar .....	5
Scrum Master Seminar .....	5
Agile Coach Seminar (in Vorbereitung) .....	5
Prüfungsinhalte Übersicht .....	6
Level der Prüfungsfragen.....	6
Literaturliste/Referenzmaterial .....	7
Themen und Referenzmaterial .....	7
Themenböcke und deren Gewichtung in den Zertifizierungsprüfungen .....	10
<b>Scrum Master Zertifizierung</b> .....	<b>12</b>
Empfohlene Trainingsdauer .....	12
Prüfungsformat.....	12
Abschlüsse .....	12
Voraussetzung zur Teilnahme an der Prüfung.....	12
<b>Product Owner Zertifizierung</b> .....	<b>13</b>
Empfohlene Trainingsdauer .....	13
Prüfungsformat.....	13
Abschluss.....	13
Voraussetzung zur Teilnahme an der Prüfung.....	13
<b>Agile Coach Zertifizierung (in Vorbereitung)</b> .....	<b>14</b>
Empfohlene Trainingsdauer .....	14
Prüfungsformat.....	14
Abschluss.....	14
Voraussetzung zur Teilnahme an der Prüfung.....	14
Themen und Referenzmaterial .....	14
<b>Mögliche Seminarangebote</b> .....	<b>15</b>
Kombiprüfung.....	15
Prüfungsstrategie und mögliche Abschlusslevels.....	15
<b>Anhang</b> .....	<b>16</b>
Englische Begriffe und deren Übersetzungen ins Deutsche .....	16
Glossar.....	20

## Einführung

Dieses Dokument soll Trainingsveranstaltern helfen, Lehrgänge und Trainingsunterlagen zu entwickeln, die den Anforderungen von itedas.org gerecht werden.

Das Hauptziel dieses Syllabus (Lehrplans) besteht in der Festlegung der Prüfungsthemen, -anforderungen und -details. Die Nutzer sollen bei der Entwicklung neuer, qualitativ hochwertiger Lehrgänge unterstützt werden.

Der Syllabus „itedas.org Certified Scrum Practitioner“ ist darauf ausgerichtet, den Teilnehmern das notwendige Wissen zu vermitteln, um als Mitglied in einem Scrum Team erfolgreich arbeiten zu können.

Geprüft werden die Fähigkeiten der Teilnehmer, sich an die erklärten Begriffe und Konzepte zu „erinnern“ und diese so zu „verstehen“, dass sie in der Praxis angewandt werden können.

Erfolgreiche Teilnehmer erhalten eine international anerkannte Bestätigung ihrer Kenntnisse (Zertifikat). Dieses Zertifikat wird von Certible, einer unabhängigen und international tätigen Zertifizierungsautorität, ausgestellt.

## Zielgruppe

Das Programm richtet sich an alle, die im Rahmen eines Scrum-Projektes in der Rolle als Scrum Master, Product Owner oder als Mitglied des Entwicklungsteams erfolgreich mitarbeiten wollen.

Der Syllabus trägt der Tatsache Rechnung, dass alle Ausführenden der Scrum-Rollen über das gleiche Basiswissen verfügen müssen, um Projekte nach Scrum erfolgreich umsetzen zu können. Dies spiegelt sich auch in den verschiedenen Abschlüssen der Zertifizierungsprüfung wider.

Ganz gleich, welche Rolle Sie im Scrum Team einnehmen wollen, ob Mitglied im Entwicklungsteam, Product Owner oder Scrum Master, Sie müssen die Kernelemente von Scrum verstehen und Ihrer Rolle gemäß beherrschen. Dieses Ausbildungsprogramm bietet die entsprechende Grundlage.

### Hinweis zur Versionierung des Dokuments

Die Versionierung berücksichtigt folgende Dimensionen:

- Versionierung der Lerninhalte des Syllabus (z.B.: syl\_04.00)
- Versionierung der Sprechversion des Syllabus (z.B.: de\_01.00)

Unterschiedliche Sprachversionen (z.B.: de\_01.00, en\_02.00) mit identischer Syllabus-Version (z.B.: syl\_04.00) besitzen die gleichen Lerninhalte.

Damit ist sichergestellt, dass die Syllabus-Version in allen Sprachvarianten identisch ist und Verwechslungen ausgeschlossen werden können.

## Versionshistorie

Version	Datum	Kommentar	Autor(en)
04.00 de 00.00		überarbeitete Version auf Basis des Scrum Guide 2020 und aktueller Strömungen in der Praxis; Ersatz der Blooms-Level-Taxierung (sechs Level) durch ein einfacheres System mit drei Leveln (siehe Level der Prüfungsfragen)	Gerd Bauer Dierk Söllner Hans-Jörg Steffe
03.02 de 01.00	01.02.2020	<ul style="list-style-type: none"> <li>• Sprachversionierung eingefügt</li> <li>• ISBN für die deutsche Version des Buches „Kenneth S. Rubin: Essential Scrum. mitp 2014“ (siehe Literaturliste/Referenzmaterial #2) angegeben</li> <li>• Fehler in der Referenzliste behoben: die Buchreferenzen 02 und 04 waren vertauscht</li> </ul>	Gerd Bauer
03.01 de 01.00	10.10.2019	<ul style="list-style-type: none"> <li>• Product-Owner-Zertifizierung ohne Voraussetzung (Scrum-Master-Zertifizierung) möglich</li> <li>• Kombiprüfung: Scrum Master und Product Owner</li> </ul>	Gerd Bauer
03.00 de 01.00	23.07.2019	initiale Erstellung	Gerd Bauer
vor 03.00 de	–	unveröffentlichte Projektversionen	Gerd Bauer

## Certified Scrum Practitioner (CSP) – Curriculum Überblick

Das CSP-Curriculum bietet drei auf die Praxis abgestimmte Zertifizierungslevel:

- Product Owner
- Scrum Master
- Agile Coach (in Vorbereitung)



### Product-Owner-Seminar

Neben der Vermittlung des grundlegenden Wissens über Scrum stehen hier die Aufgaben und Verantwortlichkeiten der Rolle des Product Owners im Mittelpunkt. Folglich liegt der Schwerpunkt der Seminarinhalte bei der agilen Produktplanung, dem Erstellen und der Pflege des Product Backlogs sowie beim Schreiben und Vermitteln von User Stories nach dem 3C-Prinzip.

### Scrum-Master-Seminar

Neben der Vermittlung des grundlegenden Wissens über Scrum stehen hier die Aufgaben und Verantwortlichkeiten der Rolle des Scrum Masters im Mittelpunkt. Als dienende Führungskraft muss der Scrum Master nicht nur die Aufgaben und Verantwortlichkeiten der anderen Rollen verstehen und vermitteln können. Darüber hinaus muss der Rolleninhaber auch erklären können, warum welche Regeln in Scrum gelten.

### Agile-Coach-Seminar (in Vorbereitung)

## Prüfungsinhalte Übersicht

### Level der Prüfungsfragen

Der Schwierigkeitsgrad der Prüfungsfragen zu den einzelnen Themenblöcken im Syllabus (Syllabus-ID; SID) orientiert sich an dem jeweiligen Zertifizierungslevel (Scrum Master, Product Owner oder Agile Coach) und gliedert sich in folgende Stufen (Knowledge Level; KL):

- **KL1: Kennen**

Zur Beantwortung der Fragen in diesem Level ist es ausreichend, sich an den gelernten Stoff erinnern und ihn abrufen zu können.

Beispiel: Welche der folgenden Rollen ist in Scrum nicht definiert?

- A. Scrum Master
- B. Developer
- C. Product Owner
- D. Project Manager

- **KL2: Verstehen**

Zur Beantwortung der Fragen in diesem Level muss auf Basis des gelernten Stoffs die Bedeutung von gegebenem Inhalt oder Situationen erkannt werden.

Beispiel: Als Scrum Master erfahren Sie von den Developern, dass diese den Sprint abgebrochen haben, um sich vorrangig um den Abbau von technischen Schulden (Technical Debt) zu kümmern. Wie schätzen Sie diese Situation ein?

- A. Sie erkennen, dass die Developer sich engagiert einbringen und agil auf eine erkannte Situation reagieren.
- B. Sie erkennen an der Reaktion der Developer das Handeln nach dem Werteprinzip „Reagieren auf Veränderungen ist wichtiger als das Befolgen eines Plans“.
- C. Sie erkennen an der Reaktion der Developer das Handeln nach dem Prinzip „In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann, und passt sein Verhalten entsprechend an“.
- D. Sie erkennen, dass die Developer offensichtlich nicht die Zuständigkeiten im Scrum Team beachtet haben.

- **KL3: Anwenden**

Zur Beantwortung der Fragen in diesem Level muss auf Basis der erkannten Bedeutung einer Situation Wissen richtig angewendet werden.

Beispiel: Als Scrum Master erfahren Sie von den Developern, dass diese den Sprint abbrechen werden, um sich vorrangig um den Abbau von technischen Schulden (Technical Debt) zu kümmern. Wie sollten Sie reagieren?

- A. Wenn Ihnen die Begründung einleuchtet, stimmen Sie der Entscheidung zu, was zum Abbruch des Sprints führt.
- B. Weil die Developer die Dringlichkeit der Behebung von technischen Schulden am besten beurteilen können, stimmen Sie der Entscheidung zu.
- C. Wenn Ihnen die Begründung einleuchtet, reden Sie mit dem Product Owner und veranlassen in Absprache mit dem Product Owner den Abbruch des Sprints.
- D. Sie veranlassen ein Meeting, in dem sich die Developer mit dem Product Owner austauschen sollen, damit der Product Owner über den Sprint-Abbruch entscheiden kann.

## Literaturliste/Referenzmaterial

ID	Quelle
1	<b>Ken Schwaber, Jeff Sutherland: Scrum Guide German Nov 2020</b> <a href="https://scrumguides.org/download.html">https://scrumguides.org/download.html</a>
2	Kenneth S. Rubin: Essential Scrum. mitp 2014 ISBN-13: 978-0137043293
3	Mike Cohn: Succeeding with Agile. Addison Wesley 2010 ISBN-13: 978-0-321-57936-2
4	Handbuch über RE@Agile nach dem IREB-Standard Version 2.0 Dezember 2019
5	<a href="http://agilemanifesto.org/iso/de/manifesto.html">http://agilemanifesto.org/iso/de/manifesto.html</a>
6	Ken Schwaber: Nexus Guide German 2021, Quelle: <a href="https://www.scrum.org/resources/nexus-guide">https://www.scrum.org/resources/nexus-guide</a>
7	SAFe® 5 Achieving Business Agility with SAFe 5, A Scaled Agile, Inc. White Paper February 2021; Quelle: <a href="https://www.scaledagile.com/resources/safe-whitepaper/">https://www.scaledagile.com/resources/safe-whitepaper/</a>
8	<a href="https://less.works/de">https://less.works/de</a>
9	<a href="https://www.scruminc.com/the-3-5-3-of-scrum/">https://www.scruminc.com/the-3-5-3-of-scrum/</a>
10	<a href="https://www.atlassian.com/de/agile/agile-at-scale/agile-iron-triangle">https://www.atlassian.com/de/agile/agile-at-scale/agile-iron-triangle</a>
11	<a href="https://en.wikipedia.org/wiki/Test_automation">https://en.wikipedia.org/wiki/Test_automation</a>

## Themen und Referenzmaterial

Der Scrum Guide 2020 bildet DIE verbindliche Grundlage für **alle** Themenbereiche (Syllabus-IDs) und ist daher in den Referenzlisten nicht zusätzlich aufgeführt. Sollten die angegebenen zusätzlichen Quellen etwas anderes aussagen als der Scrum Guide 2020, **so gelten für die Prüfungen immer die Aussagen des Scrum Guide als richtig.**

## Scrum Master und Product Owner

Syllabus-ID	Kapitel	Quellen (siehe Literaturliste)
<b>01.00</b>	<b>Agile Theorie und Prinzipien</b>	
01.01	Klassische vs. agile Herangehensweise	2: Kap.: 3
01.02	Iterationen	2: Kap.: 3
01.03	Das agile Manifest und die Prinzipien	5
01.04	Agile Methoden und Prozesse	4
01.05	Scrum-Theorie, Werte und Prinzipien	2: Kap.: 3 alle Unterkapitel
01.06	Einsatz von Scrum (Wann ist Scrum gut geeignet?)	2: Kap.: 1
01.07	Die drei Säulen von Scrum	
01.08	Transparenz	
01.09	Die 3x5x3-Regel	

<b>02.00</b>	<b>Die Scrum-Rollen</b>	
02.01	Ein Scrum Team	
02.02	Developer	
02.03	Product Owner	
02.04	Scrum Master	
<b>03.00</b>	<b>Die Produktplanung</b>	
03.01	Planungsprinzipien	2: Kap.: 14 alle Unterkapitel
03.02	Planung auf mehreren Ebenen	2: Kap.: 15 alle Unterkapitel
03.03	Der Start in Scrum (Start-up im Team)	4
03.04	Entwicklung einer Vision	2: Kap.: 17, 17.1 alle Unterkapitel, 17.3
03.05	Das Produktziel	
03.06	Die Scrum-Artefakte im Überblick	
03.07	Das Product Backlog	2: Kap.: 6.1, 6.2
03.08	Deep	2: Kap.: 6.3
03.09	Der Refinement-Prozess	2: Kap.: 6.4
03.10	Die Granularität von Anforderungen	4
03.11	User Storys (inkl. 3C)	2: Kap.: 5 alle Unterkapitel (außer 5.6)
03.12	INVEST	2: Kap.: 5.6
03.13	Die Priorisierung von Anforderungen	4
03.14	Die Aufwandsschätzung	2: Kap.: 7 alle Unterkapitel
03.15	Definition of Ready	2: Kap.: 6.5
<b>04.00</b>	<b>Der Scrum-Prozess (Aktivitäten und Artefakte)</b>	
04.01	Sprint	2: Kap.: 20.1, 20.1.1–3, Abb. 20.2, Abb. 20.4, 20.3.3, 20.5, Abb. 20.5
04.02	Sprint-Abbruch	2: Kap. 4.5.6
04.03	Sprint Planning	2: Kap.: 19 alle Unterkapitel
04.04	Velocity	2: Kap.: 7.6, 7.7, 7.8, 7.9, 7.10
04.05	Sprint Backlog	
04.06	Sprint-Ziel	2: Kap.: 19.6
04.07	Task Board	2: Kap.: 20.6.1
04.08	Daily Scrum	2: Kap.: 2.3.5
04.09	Sprint Review	2: Kap.: 21 alle Unterkapitel
04.10	Increment	2: Kap.: 2.3.6, Glossar
04.11	Definition of Done	2: Kap.: 2.3.6, 4.6, 4.6.1–4
04.12	Retrospektive	2: Kap.: 22, 22.1, 22.2, 22.4, Abb. 22.4

<b>05.00</b>	<b>Steuern und Monitoren</b>	
05.01	Release-Planung	2: Kap.: 14.7, Abb. 14.4, Abb. 14.5
05.02	Potenziell auslieferbares Produkt	4
05.03	MVP/MMP	2: Seite 338
05.04	Permanente Wertsteigerung	4
05.05	Scope fixed vs. Time fixed	10
05.06	Burn down/up	2: Kap.: 20.6.2–3
<b>06.00</b>	<b>Qualitätskriterien</b>	
06.01	Automatisiertes Testen	11
06.02	Technische Schulden	2: Kap.: 8, 8.1, 8.2, 8.3, 8.4, Abb. 8.7
<b>07.00</b>	<b>Skalierbarkeit von Scrum</b>	
07.01	Frameworks (SAFe, LESS, Nexus)	6, 7 und 8

### Agile Coach (in Vorbereitung)

## Themenblöcke und ihre Gewichtung in den Zertifizierungsprüfungen

Die nachfolgende Tabelle enthält einen Überblick der Gewichtung der einzelnen Themen in den jeweiligen Zertifizierungsprüfungen und des in der Prüfung maximal geforderten Schwierigkeitsgrades der Fragen entsprechend dem Knowledge Level.

### Scrum Master und Product Owner

neue SID	Kapitel	Scrum Master		Product Owner	
		Knowledge Level	Gewichtung	Knowledge Level	Gewichtung
<b>01.00</b>	<b>Agile Theorie und Prinzipien</b>		<b>12%</b>		<b>12%</b>
01.01	Klassische vs. agile Herangehensweise	KL1: Kennen		KL1: Kennen	
01.02	Iterationen	KL1: Kennen		KL2: Verstehen	
01.03	Das agile Manifest und die Prinzipien	KL2: Verstehen		KL2: Verstehen	
01.04	Agile Methoden und Prozesse	KL1: Kennen		KL1: Kennen	
01.05	Scrum-Theorie, Werte und Prinzipien	KL2: Verstehen		KL2: Verstehen	
01.06	Einsatz von Scrum (Wann ist Scrum gut geeignet?)	KL2: Verstehen		KL2: Verstehen	
01.07	Die drei Säulen von Scrum	KL2: Verstehen		KL2: Verstehen	
01.08	Transparenz	KL2: Verstehen		KL3: Anwenden	
01.09	Die 3x5x3-Regel	KL2: Verstehen		KL2: Verstehen	
<b>02.00</b>	<b>Die Scrum-Rollen</b>		<b>12%</b>		<b>12%</b>
02.01	Ein Scrum Team	KL3: Anwenden		KL2: Verstehen	
02.02	Developer	KL2: Verstehen		KL2: Verstehen	
02.03	Product Owner	KL2: Verstehen		KL3: Anwenden	
02.04	Scrum Master	KL3: Anwenden		KL2: Verstehen	
<b>03.00</b>	<b>Die Produktplanung</b>		<b>27%</b>		<b>38%</b>
03.01	Planungsprinzipien	KL1: Kennen		KL3: Anwenden	
03.02	Planung auf mehreren Ebenen	KL1: Kennen		KL3: Anwenden	
03.03	Der Start in Scrum (Start-up im Team)	KL2: Verstehen		KL2: Verstehen	
03.04	Entwicklung einer Vision	KL1: Kennen		KL3: Anwenden	
03.05	Das Produktziel	KL1: Kennen		KL3: Anwenden	
03.06	Die Scrum-Artefakte im Überblick	KL2: Verstehen		KL2: Verstehen	
03.07	Das Product Backlog	KL2: Verstehen		KL3: Anwenden	
03.08	Deep	KL2: Verstehen		KL3: Anwenden	
03.09	Der Refinement-Prozess	KL1: Kennen		KL3: Anwenden	
03.10	Die Granularität von Anforderungen	KL2: Verstehen		KL3: Anwenden	
03.11	User Storys (inkl. 3C)	KL3: Anwenden		KL3: Anwenden	
03.12	INVEST	KL3: Anwenden		KL3: Anwenden	
03.13	Die Priorisierung von Anforderungen	KL1: Kennen		KL3: Anwenden	
03.14	Die Aufwandsschätzung	KL3: Anwenden		KL2: Verstehen	
03.15	Definition of Ready	KL3: Anwenden		KL2: Verstehen	

<b>04.00</b>	<b>Der Scrum-Prozess (Aktivitäten und Artefakte)</b>	<b>38%</b>	<b>24%</b>
04.01	Sprint	KL3: Anwenden	KL2: Verstehen
04.02	Sprint-Abbruch	KL3: Anwenden	KL2: Verstehen
04.03	Sprint Planning	KL3: Anwenden	KL3: Anwenden
04.04	Velocity	KL2: Verstehen	KL2: Verstehen
04.05	Sprint Backlog	KL3: Anwenden	KL1: Kennen
04.06	Sprint-Ziel	KL3: Anwenden	KL2: Verstehen
04.07	Task Board	KL3: Anwenden	KL1: Kennen
04.08	Daily Scrum	KL3: Anwenden	KL1: Kennen
04.09	Sprint Review	KL3: Anwenden	KL3: Anwenden
04.10	Increment	KL3: Anwenden	KL2: Verstehen
04.11	Definition of Done	KL3: Anwenden	KL2: Verstehen
04.12	Retrospektive	KL3: Anwenden	KL2: Verstehen
<b>05.00</b>	<b>Steuern und Monitoren</b>	<b>7%</b>	<b>10%</b>
05.01	Release-Planung	KL1: Kennen	KL3: Anwenden
05.02	Potenziell auslieferbares Produkt	KL2: Verstehen	KL2: Verstehen
05.03	MVP/MMP	KL2: Verstehen	KL2: Verstehen
05.04	Permanente Wertsteigerung	KL2: Verstehen	KL2: Verstehen
05.05	Scope fixed vs. Time fixed	KL2: Verstehen	KL2: Verstehen
05.06	Burn down/up	KL1: Kennen	KL1: Kennen
<b>06.00</b>	<b>Qualitätskriterien</b>	<b>2%</b>	<b>2%</b>
06.01	Automatisiertes Testen	KL2: Verstehen	KL1: Kennen
06.02	Technische Schulden	KL2: Verstehen	KL2: Verstehen
<b>07.00</b>	<b>Skalierbarkeit von Scrum</b>	<b>2%</b>	<b>2%</b>
07.01	Frameworks (SAFe, LESS, Nexus)	KL1: Kennen	KL1: Kennen

### Agile Coach (in Vorbereitung)

## Scrum-Master-Zertifizierung



Certified  
Scrum Practitioner  
**Scrum Master**



### Empfohlene Trainingsdauer

2 Tage bzw. 16 Unterrichtseinheiten à 0:45 Stunden. Ideal 3 Tage bzw. 24 Unterrichtseinheiten mit mehr praktischen Übungen.

### Prüfungsformat

<b>Art:</b>	Multiple-Choice-Prüfung
<b>Anzahl Fragen:</b>	40
<b>Antwortmöglichkeiten pro Frage:</b>	4
<b>Richtige Antworten pro Frage:</b>	1
<b>Dauer:</b>	60 Minuten zzgl. 15 Minuten für nachgewiesenen Nachteilsausgleich (andere Muttersprache als Prüfungssprache, nachgewiesene relevante Krankheiten/Behinderungen)

### Abschlüsse

Richtige Antworten	Score	Erreichtes Zertifizierungslevel
0–20	0 %–50 %	keines
21–29	> 50 %–< 75 %	Scrum Foundation
30–40	75 %–100 %	Scrum Master

### Voraussetzung zur Teilnahme an der Prüfung

- keine
- Die Teilnahme an einem Seminar wird empfohlen, damit die in der Prüfung verwendete Begriffswelt (siehe Literaturliste) ausreichend vertraut ist.

## Product-Owner-Zertifizierung



Certified  
Scrum Practitioner  
Product Owner



### Empfohlene Trainingsdauer

2 Tage bzw. 16 Unterrichtseinheiten à 0:45 Stunden. Ideal 3 Tage bzw. 24 Unterrichtseinheiten mit mehr praktischen Übungen.

### Prüfungsformat

Art:	Multiple-Choice-Prüfung
Anzahl Fragen:	40
Antwortmöglichkeiten pro Frage:	4
Richtige Antworten pro Frage:	1
Dauer:	60 Minuten zzgl. 15 Minuten für nachgewiesenen Nachteilsausgleich (andere Muttersprache als Prüfungssprache, nachgewiesene relevante Krankheiten/Behinderungen)

### Abschluss

Richtige Antworten	Score	Erreichtes Zertifizierungslevel
0–20	0 %–50 %	keines
21–29	> 50 %–< 75 %	Scrum Foundation
30–40	75 %–100 %	Product Owner

### Voraussetzung zur Teilnahme an der Prüfung

- keine
- Die Teilnahme an einem Seminar wird empfohlen, damit die in der Prüfung verwendete Begriffswelt (siehe Literaturliste) ausreichend vertraut ist.

## Agile-Coach-Zertifizierung (in Vorbereitung)



Certified  
Scrum Practitioner  
**Agile Coach**



### Empfohlene Trainingsdauer

### Prüfungsformat

Art:	
Anzahl Fragen:	
Antwortmöglichkeiten pro Frage:	
Richtige Antworten pro Frage:	
Dauer:	

### Abschluss

Richtige Antworten	Score	Erreichtes Zertifizierungslevel
		keines
		Agile Coach

### Voraussetzung zur Teilnahme an der Prüfung

-  Scrum-Master- UND Product-Owner-Zertifizierung von itedas.org, scrum.org, ScrumAlliance oder scruminc
- Die Teilnahme an einem Seminar wird empfohlen, damit die in der Prüfung verwendete Begriffswelt (siehe Literaturliste) ausreichend vertraut ist.

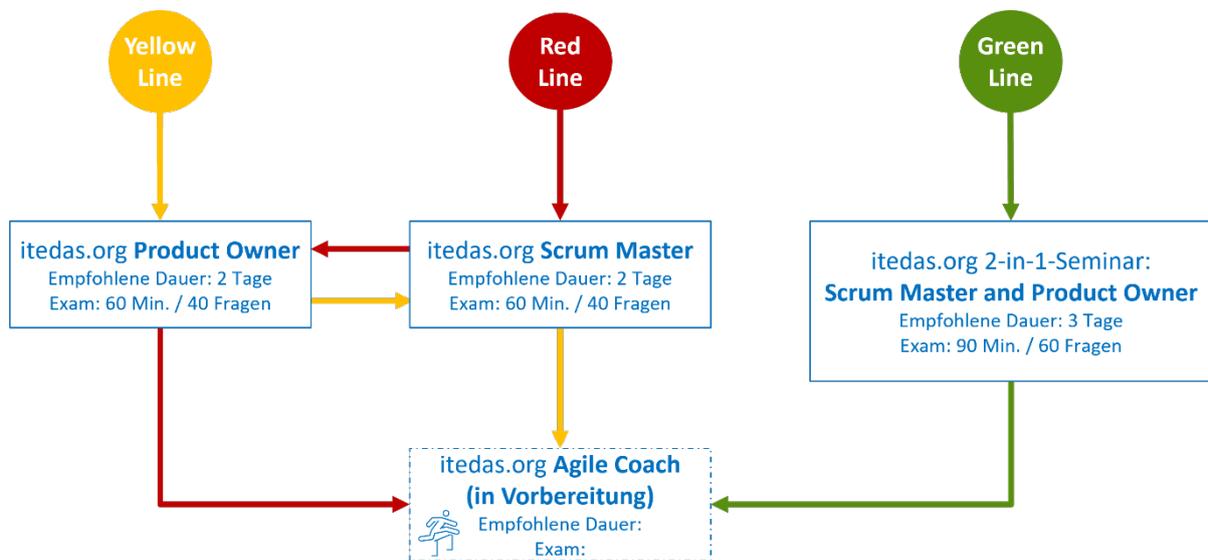
### Themen und Referenzmaterial

## Mögliche Seminarangebote

Wie dargestellt, bauen die Seminare aufeinander auf. Um den Trainern die Möglichkeit zu geben, den Veranstaltungsverlauf so individuell wie möglich auf die Kundenbedürfnisse anpassen zu können, sind ab dieser Version des Syllabus die Prüfungsinhalte der einzelnen Zertifizierungsprüfungen so aufeinander abgestimmt, dass die einzelnen Qualifikationslevel möglichst frei kombinierbar sind und mit einer Kombi-Zertifizierungsprüfung abgeschlossen werden können.

Die Prüfungsfragen der Kombi-Prüfungen entsprechen den Prüfungsfragen der Einzel-Zertifizierungsprüfungen. Damit ist bei der Auswertung der Fragebögen sichergestellt, dass der Prüfling erst dann ein Zertifikat für einen höheren Abschluss erhält, wenn die Voraussetzungen dafür erfüllt sind.

Aufgrund dieser Gestaltung haben die Trainer Freiraum zur Gestaltung von maßgeschneiderten Veranstaltungen. Folgende Seminarreihen (Yellow, Red und Green Line) werden empfohlen:



## Kombi-Prüfung

Art:	Multiple-Choice-Prüfung
Anzahl Fragen:	60
Antwortmöglichkeiten pro Frage:	4
Richtige Antworten pro Frage:	1
Dauer:	90 Minuten zzgl. 25 Minuten für nachgewiesenen Nachteilsausgleich (andere Muttersprache als Prüfungssprache, nachgewiesene relevante Krankheiten/Behinderungen)

## Prüfungsstrategie und mögliche Abschlusslevels

Erreichte Punkte	Score	Zertifizierungslevel
0–30	0–50 %	keines
31–44	> 50 %–< 75 %	Scrum Foundation
45–60	75 %–100 %	Scrum Master & Product Owner

## Anhang

### Englische Begriffe und ihre Übersetzungen ins Deutsche

Englisch	Deutsch
<b>3</b>	
353 Rule	353-Regel
3C	3C
<b>A</b>	
Acceptance Conditions/Criteria	Akzeptanzkriterien
Acceptance Test	Akzeptanztest
Adapt	Anpassung
ADAPT	ADAPT
Affinity Estimation	Affinity Estimation
Agile Manifesto	agiles Manifest
Artefact	Artefakt
<b>B</b>	
Backlog Item	Backlog Item
Burn-down Chart	Burn-down-Chart
Burn-up Chart	Burn-up-Chart
Bus Factor	Bus-Faktor
Business Value	Geschäftswert
Business Value Chart	Business-Value-Chart
<b>C</b>	
Cadence	Kadenz
Collective Code Ownership	gemeinsamer Code-Besitz (Collective Code Ownership)
Constraints (Requirements)	Randbedingungen (Anforderungen)
Cynefin Framework	Cynefin-Framework
<b>D</b>	
Daily Scrum	Daily Scrum
Daily Stand-up	Daily Stand-up
DEEP	DEEP
Defects	Fehler
Definition of Done (DoD)	Definition of Done (DoD)
Definition of Ready (DoR)	Definition of Ready (DoR)
Developer	Developer
Development Team	Entwicklungsteam
<b>E</b>	
Empirical Process Control	empirische Prozesssteuerung

Enterprise Transition Community (ETC)	Enterprise Transition Community (ETC)
Epic	Epic
Escaped Defects	nicht entdeckte Fehler
Estimation	Schätzung
Estimation Concept	Schätzkonzept
Estimation Unit	Schätzeinheit
<b>F</b>	
Framework	Rahmenwerk
Function Requirements	funktionale Anforderungen
<b>I</b>	
Ideal Days	Idealtage
Ideal Hours	Idealstunden
Ideal Time	Idealzeit
Impediment	Hemmnis (Impediment)
Increment	Inkrement
Information Radiator	Information Radiator
Inspect	Überprüfung (Inspect)
INVEST	INVEST
<b>K</b>	
Kanban Board	Kanban Board
Kano Model	Kano-Modell
Kinds of Estimation	Schätzarten
<b>L</b>	
Lean	Lean
<b>M</b>	
Magic Estimation	Magic Estimation
Minimum Viable Product (MVP)	Minimum Viable Product (MVP)
Minimum Marketable Product (MMP)	Minimum Marketable Product (MMP)
<b>N</b>	
Nexus	Nexus
Niko-Niko Calendar	Niko-Niko-Kalender
<b>O</b>	
Osmotic Communication	osmotische Kommunikation
<b>P</b>	
PDCA Approach / PDCA Cycle	PDCA-Ansatz / PDCA-Zyklus
Planning Onion	Planning Onion
Planning Poker	Planning Poker
Portfolio	Portfolio

Portfolio Backlog	Portfolio-Backlog
Product	Produkt
Product Backlog	Product Backlog
Product Backlog Refinement	Product Backlog Refinement
Product Goal	Produktziel
Product Owner	Product Owner
Product Planning	Produktplanung
Product Vision	Produktvision
<b>Q</b>	
Quality Gate	Quality Gate
<b>R</b>	
Refinement Meeting	Refinement-Meeting
Release	Release
Release Plan	Release-Plan
Release Train	Release Train
Requirements	Anforderungen/Requirements
Retrospective	Retrospective
Return on Investment (ROI)	Return on Investment (ROI)
<b>S</b>	
SAFe	SAFe
Scaled Agile Framework (SAFe)	Scaled Agile Framework (SAFe)
Scaling Scrum	Skalieren von Scrum
Scrum Master	Scrum Master
Scrum of Scrums	Scrum of Scrums
Scrum Planning Principles	Scrum-Planungsprinzipien
Servant Leader	dienende Führungskraft
Sprint	Sprint
Sprint Backlog	Sprint Backlog
Sprint Burn-down Chart	Sprint-Burn-down-Chart
Sprint Goal	Sprint-Ziel
Sprint Planning	Sprint-Planning
Sprint Retrospective	Sprint-Retrospective
Sprint Review	Sprint-Review
Stakeholder	Stakeholder
Stand-up Meeting	Stand-up-Meeting
Story Mapping	Story Mapping
Story Points	Story Points
<b>T</b>	
Task Board	Task Board
Technical Debt	technische Schulden

Test-Driven Development (TDD)	Test-Driven Development (TDD)
Tests	Tests
Theme	Thema
Time Box	Timebox
Time2Market	Time2Market
Triangulation	Triangulation
True Leader	True Leader
<b>U</b>	
User Story	User Story
<b>V</b>	
Velocity	Velocity
<b>W</b>	
WIP-Limit	WIP-Limit
Work in Progress (WIP)	Work in Progress (WIP)

## Glossar

Zum Verstehen der Prüfungsfragen in der Zertifizierungsprüfung sollten folgende Begriffe bekannt sein und eingeordnet werden können. In Klammer stehen die englischen Begriffe.

**353-Regel** Scrum beschreibt drei Rollen, fünf Events (Ereignisse) und drei Artefakte.

**3C** Bei der Erstellung und Kommunikation der Anforderungen einer User Story sollten folgende 3 Cs beachtet werden:

- **Card:** Karte mit der dokumentierten User Story sowie ggf. mit einigen weiteren Stichworten
- **Conversation** (Gespräch): Erfassung und Weitergabe der User Story erfolgen in einem regen Austausch zwischen Stakeholder, Product Owner und Developer.
- **Confirmation:** Zu jeder User Story gehören eindeutige Testszenarien, anhand derer der mit ihr beabsichtigte Nutzen getestet werden kann.

**ADAPT** Das Akronym ADAPT fasst die fünf Schritte zur erfolgreichen Einführung agiler Vorgehensweisen zusammen:

**A**wareness – **D**esire – **A**bility – **P**romotion – **T**ransfer

- **Awareness** meint das Bewusstsein, dass der derzeit verwendete Prozess nicht die gewünschten Ergebnisse bringt.
- **Desire** beschreibt den Willen, die bestehenden Probleme mithilfe agiler Methoden zu adressieren.
- **Ability** ist die Befähigung, eine agile Methode erfolgreich einzusetzen.
- **Promotion** meint eine Bekanntmachung im Sinne von (internem) Marketing für die agile Vorgehensweise.
- **Transfer** ist die Ausweitung der agilen Methoden auf andere Unternehmensbereiche.

**Affinity Estimation** Bei der Affinity Estimation werden die Backlog-Elemente ihrem geschätzten Aufwand nach geordnet. Elemente mit geringem Aufwand werden links angeordnet, solche mit größerem werden rechts angeordnet. Für jedes Element wird bestimmt, ob dieses größer, kleiner oder gleich umfangreich wie jedes andere Element ist, wodurch sich eine nach dem Aufwand sortierte Reihenfolge ergibt.

**agiles Manifest (Agile Manifesto)** Das agile Manifest ([agilemanifesto.org/iso/de/manifesto.html](http://agilemanifesto.org/iso/de/manifesto.html)) ist im Jahre 2001 durch die Vertreter der führenden agilen Methoden gemeinsam entwickelt und formuliert worden. Es bildet die Basis für ein Verständnis von „Agilität“ und besteht aus den 4 Kernsätzen sowie 12 Prinzipien zu Konkretisierung und besserem Verständnis.

**Akzeptanzkriterien (Acceptance Conditions/Criteria)** Akzeptanzkriterien beziehen sich auf einzelne konkrete Product Backlog Items und helfen, objektiv und eindeutig zu prüfen, ob eine Funktionalität umgesetzt wurde.

**Akzeptanztest (Acceptance Test)** Ein Akzeptanztest wird aus den Akzeptanzkriterien abgeleitet. Er überprüft, ob ein Feature seine geforderte Funktion erfüllt und hat in der Regel nur zwei mögliche Ergebnisse: „erfüllt“ oder „nicht erfüllt“.

**Anforderungen/Requirements** Von einem Produkt zu erfüllende Aspekte. Anforderungen an ein Produkt können in funktionale Anforderungen, Randbedingungen und Qualitätsanforderungen unterschieden werden.

**Anpassung (Adapt)** siehe Scrum Guide 2020

**Artefakt (Artefact)** siehe Scrum Guide 2020

**Backlog Item** siehe Scrum Guide 2020

**Burn-down-Chart** Grafische Darstellung der, für ein bestimmtes Zeitfenster, noch verbleibenden Arbeit.

**Burn-up-Chart** Das Burn-up-Chart zeigt, welche (wie viel) Arbeit bereits abgeschlossen ist und wie viel Arbeit noch offen ist, anhand eines Flächen- oder Liniendiagramms. Da in Scrum (in der Agilität generell) davon ausgegangen wird, dass zu jeder Zeit neue oder geänderte Anforderungen von Stakeholdern eintreffen können, ist diese Form der Darstellung eine in der Agilität häufig verwendete. Das Burn-up-Chart geht dabei eigentlich nicht davon aus, dass die Arbeit irgendwann abgeschlossen sein wird. Es wird vielmehr oftmals in Verbindung

mit dem erwarteten Business Value pro Anforderung in Verbindung gebracht. Sollte der Business Value pro Anforderung nicht mehr hoch genug sein, um Mehrwert zu schaffen, wird die Weiterentwicklung eingestellt. Auch aus diesem Grund findet das Burn-up-Chart eher in Produktentwicklungen seinen Einsatz, bei denen nicht im Vorhinein ein fixer Endzeitpunkt vorgesehen ist.

**Bus-Faktor (Bus Factor)** Der Bus-Faktor ist ein Maß für das Risiko, das sich dadurch ergibt, dass Informationen in einem Team nicht ausreichend geteilt wurden bzw. dass benötigte Fähigkeiten in einem Team nicht ausreichend vorhanden sind. Ein Bus-Faktor von eins bedeutet, dass beim Ausfall des betreffenden Teammitglieds (also, falls es vom Bus überfahren wird) die Arbeit ins Stocken gerät. (Quelle: Wikipedia)

**Business-Value-Chart** Im Business-Value-Chart wird der mit jedem Sprint (oder auch Release) zu erwartende Business-Nutzen dargestellt.

**Cynefin-Framework** Das Cynefin-Framework ist ein Modell zur Einordnung von Aufgaben, Problemen und Situationen. Es unterstützt primär eine Entscheidungsfindung. Cynefin liefert eine typologische Einordnung von Systemen und besteht aus 5 Feldern, in die sich Systeme, Probleme und Themen einordnen lassen.

1. Bei „**Clear**“ ist die Beziehung zwischen Ursache und Wirkung für alle offensichtlich.
2. Bei „**Complicated**“ ist eine Analyse oder andere Form der Anwendung von Fachwissen erforderlich, um die Beziehung zwischen Ursache und Wirkung herzustellen.
3. Bei „**Complex**“ kann die Beziehung zwischen Ursache und Wirkung erst im Nachhinein wahrgenommen werden.
4. Bei „**Chaotic**“ gibt es auf Systemebene keine Beziehung zwischen Ursache und Wirkung (wohl aber in der Systemumwelt, jedoch nicht erfahrbare).
5. „**Disorder**“ ist der Zustand des Nichtwissens, welche Art von Kausalität besteht.

**Daily Scrum (Daily Scrum)** siehe Scrum Guide 2020

**Daily Stand-up (Daily Stand-up)** siehe Daily Scrum

**DEEP (DEEP)** DEEP ist ein Akronym, das eine Gruppe von Kriterien bezeichnet, mit denen sich die Qualität eines Product Backlogs beurteilen lässt. Diese Kriterien sind:

- **D**etailed appropriately (ausreichend detailliert)
- **E**mergent (neu auftretend, entstehend, sich entwickelnd)
- **E**stimated (geschätzt)
- **P**rioritized (priorisiert)

**Definition of Done (DoD) (Definition of Done (DoD))** Die Definition of Done ist eine Sammlung von Kriterien, die definiert, wann die Arbeit an einem Product Increment vollständig abgeschlossen (fertig) ist, sodass das Produkt als potenziell auslieferbar gilt. Diese Definition of Done muss sowohl vom gesamten Increment als auch von jedem einzelnen Teil des Increments erfüllt sein.

**Definition of Ready (DoR) (Definition of Ready (DoR))** Die Definition of Ready ist eine Sammlung von Kriterien, die definiert, wann ein Product-Backlog-Element bereit ist, um in die Sprint-Planung mit aufgenommen zu werden. Das Element ist also ausreichend genug definiert, um von den Entwicklern in konkrete Tasks (Aufgaben) zerlegt werden zu können.

**Developer** siehe Scrum Guide 2020

**dienende Führungskraft (Servant Leader)** Der Servant-Leadership-Ansatz beinhaltet eine kompromisslose Ausrichtung der Führung auf die Interessen der Geführten: „Ein Servant Leader liebt Menschen und möchte ihnen helfen. Die Mission des Servant Leaders ist es daher, die Bedürfnisse anderer zu identifizieren und zu versuchen, diese Bedürfnisse zu befriedigen.“ (Kent Keith, CEO des Greenleaf Center for Servant Leadership)

**empirische Prozesssteuerung (Empirical Process Control)** siehe Scrum Guide 2020

**Enterprise Transition Community (ETC)** Eine Enterprise Transition Community (kurz: ETC) ist ein Scrum Team, das die Transition hin zu agilem Vorgehen in einem Unternehmen durchführt.

**Epic (Epic)** Anforderungen, die noch unklar und/oder noch sehr groß sind, werden als Epic bezeichnet. In der Praxis können Epics als Zusammenfassung mehrerer Anforderungen (Features, User Stories, ...) genutzt werden. Ein Epic ist in der Regel nicht in einem Sprint umsetzbar.

**Fehler (Defects)** Alles in einem Produkt, was vom Scrum Team als „falsch“ angesehen wird; Fehler führen normalerweise dazu, dass ein neues Element zum Backlog hinzugefügt wird.

**funktionale Anforderungen (Function Requirements)** Anforderungen, die an das gelieferte Ergebnis einer Funktion erfüllt werden sollen. Beispiel: Das Produkt muss den Import im CSV-Format unterstützen.

**gemeinsamer Code-Besitz (Collective Code Ownership)** Gemeinsamer Code-Besitz meint, dass alle Mitglieder des Entwicklungsteams den Code auch ohne Rücksprache mit Dritten verändern dürfen.

**Geschäftswert (Business Value)** Der Geschäftswert ist der Nutzen, den ein Backlog-Element bei Fertigstellung bringt und der unter anderem neben dem geschätzten Aufwand zur Priorisierung herangezogen werden kann.

**Hemmnis (Impediment)** Ein Umstand, der die Erreichung eines Ziels blockiert oder behindert. Dieser Begriff wird vor allem für Probleme verwendet, die ein Team an der reibungslosen Arbeit für das Erreichen des Sprint-Ziels behindern.

**Idealstunden (Ideal Hours)** siehe Idealzeit

**Idealtage (Ideal Days)** siehe Idealzeit

**Idealzeit (Ideal Time)** Als Idealzeiten werden die Zeitdauern (in Tagen oder Stunden) bezeichnet, innerhalb derer vollkommen unterbrechungsfrei an einer einzigen Aufgabe gearbeitet wird. Idealzeiten stellen eine Schätzeinheit dar, mit deren Hilfe der Aufwand für Backlog-Elemente geschätzt wird. Beim Schätzen in Idealzeiten werden potenzielle Unterbrechungen der Arbeit, beispielsweise durch administrative Tätigkeiten oder sonstige Besprechungen, nicht berücksichtigt. Häufig kommt es dabei in der praktischen Umsetzung zu Missverständnissen, sodass Idealtage so gut wie nie dem tatsächlichen Aufwand entsprechen, da zu dem Entwicklungsaufwand meist auch weitere Tätigkeiten hinzukommen.

**Increment** Ein Inkrement in Scrum ist ein potenziell einsetzbares Produkt, das zum Schluss eines jeden Sprints als Ergebnis erwartet wird. Dabei enthält jedes Inkrement die Summe aller zuvor erstellten Inkremente. Das bedeutet, dass ein Inkrement nicht alleine als das gesehen werden kann, was in einem Sprint gemacht wurde, sondern auch alles beinhaltet, was in allen vorherigen Sprints gemacht wurde.

**Information Radiator** Möglichst öffentlichkeitswirksame Mittel, um Informationen zum Status des Projekts zugänglich und transparent zu machen, z. B. Burn-down-Chart; Burn-up-Chart, aktuelle Velocity, aktuelles Product Backlog, ...

**INVEST** Ein Akronym von Kriterien, mit denen die Qualität von Anforderungen (Backlog-Elementen) evaluiert werden. INVEST steht für:

- **I**ndependent (unabhängig)
- **N**egotiable (verhandelbar)
- **V**aluable (werthaltig)
- **E**stimable (schätzbar)
- **S**mall (von angemessener Größe)
- **T**estable (testbar)

**Kadenz (Cadence)** Eine Kadenz ist eine eindeutig festgelegte zeitliche Taktung. Um bei der Überprüfung zu helfen, bietet Scrum eine Kadenz in Form seiner 5 Events.

**Kanban Board** siehe Task Board

**Kano-Modell (Kano Model)** Das Kano-Modell dient dazu, Anforderungen nach den Kriterien Basisfaktoren (als selbstverständlich vorausgesetzte Anforderungen), Leistungsfaktoren (bewusst geforderte Anforderungen) und Begeisterungsfaktoren (nicht erwartete Anforderungen) zu unterteilen. Es dient als Basis dazu, wie Anforderungen erhoben und mit Priorität versehen werden können.

Darüber hinaus gibt es noch zwei weitere Faktoren:

- unerhebliche Merkmale (Sie sind sowohl bei Vorhandensein wie auch bei Fehlen ohne Belang für den Kunden. Sie können daher keine Zufriedenheit stiften, führen aber auch zu keiner Unzufriedenheit.)
- Rückweisungsmerkmale (Sie führen bei Vorhandensein zu Unzufriedenheit, beim Fehlen jedoch zu Zufriedenheit des Kunden.)

Diese zwei Merkmale finden in der Priorisierung von Anforderungen keine weitere Verwendung, da solche Anforderungen ins Backlog gar nicht erst aufgenommen werden.

**Lean** Der Begriff Lean steht als Synonym für den Begriff „Lean Management“ und bezeichnet die Gesamtheit der Denkprinzipien, Methoden und Verfahrensweisen zur effizienten Gestaltung der gesamten Wertschöpfungskette industrieller Güter.

Ziel ist es, alle Aktivitäten, die für die Wertschöpfung notwendig sind, optimal aufeinander abzustimmen und überflüssige Tätigkeiten (Verschwendung, japanisch „muda“) zu vermeiden.

**Magic Estimation** Bei der Magic Estimation handelt es sich um eine Schätzart. Dabei wird als Erstes ein Referenzelement mit einem fixen Wert versehen. Alle weiteren zu schätzenden Elemente werden anschließend im Verhältnis zu dem Referenzelement geschätzt.

**Minimal Marketable Product (MMP)** Das MMP beschreibt ein Produkt mit dem kleinstmöglichen Feature-Set, das auf die Bedürfnisse der ersten Benutzer (Innovatoren und frühe Anwender) eingeht und somit vermarktet werden kann. Studien zufolge enthalten die meisten Softwareprodukte viele Features, die niemals oder nur sehr selten genutzt werden. Daher erscheint es vernünftig, dass Sie sich auf die Features konzentrieren, die von der Mehrheit Ihrer Stakeholder geschätzt werden, und weniger gängige Features zurückstellen. (vergl. Referenzliteratur 4)

**Minimum Viable Product (MVP)** Ein Minimum Viable Product (MVP), wörtlich ein „minimal überlebensfähiges Produkt“, ist eine minimal funktionsfähige Iteration eines Produkts, das entwickelt wird, um mit minimalem Aufwand den Kunden-, Markt- oder Funktionsbedarf zu decken, um ein für die weitere Entwicklung handlungsrelevantes Feedback zu gewährleisten.

**Nexus** Nexus (lateinisch für „Verbindung, Gefüge, Nabe“) ist ein Rahmenwerk zur Skalierung von Scrum, das aus Rollen, Events, Artefakten und Regeln besteht und das die Arbeit von ungefähr 3 bis 9 Scrum Teams verbindet. Diese Teams arbeiten an einem gemeinsamen Product Backlog, um ein integriertes Inkrement zu erstellen, welches ein Ziel erfüllt.

**nicht entdeckte Fehler (Escaped Defects)** Fehler, die trotz Qualitätssicherung nicht gefunden wurden. In der Regel werden diese Probleme von Endbenutzern gefunden, nachdem ihnen die freigegebene Version zur Verfügung gestellt wurde.

**Niko-Niko-Kalender (Niko-Niko Calendar)** Das japanische Wort „niko“ bedeutet „Lächeln“; nach einem im Japanischen üblichen Muster der Wortverdopplung hat „niko-niko“ eine Bedeutung, die näher an „smiley“ liegt.

Der Niko-Niko-Kalender soll es dem Team ermöglichen, die Stimmung im Team sichtbar zu machen. Hierzu trägt jedes Teammitglied am Ende jedes Arbeitstages eine grafische Bewertung seiner Stimmung für diesen Tag in den Kalender ein. Dies kann entweder ein handgezeichnetes „Emoticon“ (Smiley) oder ein farbiger Aufkleber sein, der einem einfachen Farbcode folgt, z. B.: blau für einen schlechten Tag, rot für neutral, gelb für einen guten Tag. Im Laufe der Zeit zeigt der Niko-Niko-Kalender Muster der Veränderung der Stimmungen des Teams oder einzelner Mitglieder.

**osmotische Kommunikation (Osmotic Communication)** Ziel der osmotischen Kommunikation ist es, dass alle Teammitglieder (insbesondere die des Entwicklungsteams, da diese täglich sehr eng zusammenarbeiten) auf dem gleichen Wissensstand sind. Dazu ist es erforderlich, dass möglichst alle Teammitglieder mitbekommen, welche (wichtigen) Themen zwischen einzelnen Teammitgliedern ausgetauscht bzw. diskutiert werden. Daher sollte insbesondere das Entwicklungsteam in einem Raum zusammensitzen. Dieser Raum muss so gestaltet sein, dass er ein angenehmes Arbeiten gestattet und dass jedes Teammitglied prinzipiell in der Lage ist, sich bei Bedarf mit Fragen oder Aussagen in den Informationsfluss einbringen zu können. Auf diese Weise unterstützt die osmotische Kommunikation auch die Selbststeuerung eines Teams.

**PDCA-Ansatz/PDCA-Zyklus (PDCA approach / PDCA cycle)** Der PDCA-Ansatz (auch als Demingkreis, Deming-Rad, Shewhart Cycle oder PDCA-Zyklus bekannt) beschreibt einen iterativen vierphasigen Prozess für

Lernen und Verbesserung des US-amerikanischen Physikers Walter Andrew Shewhart. PDCA steht hierbei für das englische Plan – Do – Check – Act, was im Deutschen auch mit ‚Planen – Tun – Überprüfen – Umsetzen‘ oder ‚Planen – Umsetzen – Überprüfen – Handeln‘ übersetzt wird. Die Ursprünge des Prozesses liegen in der Qualitätssicherung.

**Planning Onion** Planung findet im Geschäftskontext auf unterschiedlichen Ebenen statt. Sie reicht von der Ebene strategischer Planung bis hinunter zur Tagesplanung. Dabei ist von zentraler Bedeutung, dass Planung auf jeder Ebene immer mit jener auf den darüberliegenden Ebenen vereinbar ist. So wird eine Firma keine Produkte entwickeln, welche ihrer Unternehmensstrategie zuwiderlaufen. Dieses Prinzip der Planung auf verschiedenen Ebenen wird oft auch in einem zwiebel förmigen Schalenmodell dargestellt. Die Schale strategischer Planung umschließt die darunterliegenden, welche im agilen Kontext oft wie folgt bezeichnet werden:

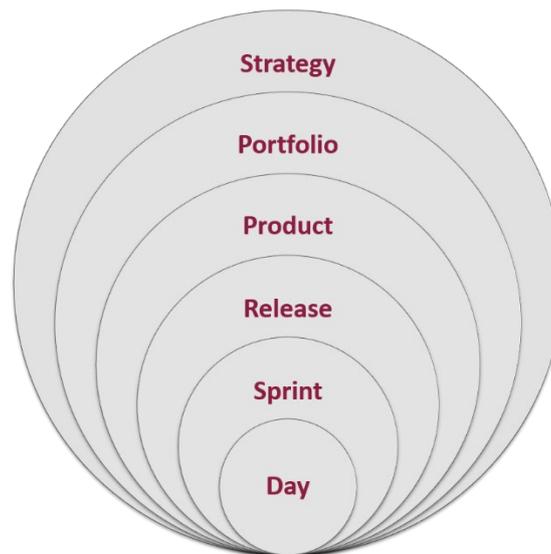


Abbildung 1: Planning Onion

Die Planning Onion ist ein sehr eingängiges Bild, um darzustellen, wie jede Planungsentscheidung auf einer Ebene sich wiederum auf die benachbarten Ebenen auswirkt. Auch hier spielen Feedback-Loops zwischen den verschiedenen Ebenen mit. Dabei ist wichtig zu verstehen, dass nicht alle Planungsebenen von denselben Personen verantwortet werden. Während die Strategie- und Portfolioplanungen meist von der Geschäftsleitung oder nahe daran angesiedelten Rollen wahrgenommen werden, fallen die darunterliegenden Ebenen zumindest teilweise in den Aufgabenbereich von Mitgliedern des Scrum Teams.

**Planning Poker** Bei der Planning-Poker-Schätzung handelt es sich um ein Schätzart. Dabei werden die Elemente mithilfe von (virtuellen) Pokerkarten, die jeweils einen bestimmten Wert aus einem festgelegten Wertevorrat (vergl. Schätzeinheiten) aufweisen, geschätzt.

**Portfolio (Portfolio)** Ein Portfolio (aus lateinisch portare, „tragen“ und folium, „Blatt“), selten Portefeuille, bezeichnet im engeren Sinne

- in einem Projekt zu erreichende Ziele und Produkte bzw.
- von einem Anbieter gebotene Produkte und Dienstleistungen (Produktportfolio).

**Portfolio-Backlog** Der Portfolio-Backlog enthält alle Anforderungen an die einzelnen Produkte im geplanten Produktportfolio und stellt somit die Summe aller Product Backlogs dar.

**Product Backlog** siehe Scrum Guide 2020

**Product Backlog Refinement** siehe Scrum Guide 2020 Refinement des Product Backlogs

**Product Owner** siehe Scrum Guide 2020

**Produkt (Product)** siehe Scrum Guide 2020

**Produktplanung (Product Planning)** Planung, mit welcher Priorität und wann welche Anforderungen eines Produktes umgesetzt werden sollen. Das Ergebnis der Produktplanung ist der Release-Plan.

Entsprechend dem Planungsprinzip „Vorabplanung sollte hilfreich, aber nicht exzessiv sein“ wird die Produktplanung bei einem agilen Vorgehen immer auf Sicht erfolgen. Grobe Anforderungen werden dabei eher längerfristig angestrebt. Detailliertere Anforderungen werden je nach angestrebten Release-Zyklen lediglich für das aktuelle und die nächsten 2 bis 3 Releases beschrieben.

Auf Basis des Release-Plans ergeben sich die Sortierung und Priorisierung der Product Backlog Items im Product Backlog.

Für die Priorisierung der Product Backlog Items kann unter anderem das Kano-Modell oder die Return-on-Investment-Methode genutzt werden.

Wie in der Planning Onion dargestellt, beruht die Produktplanung auf der Strategie des Unternehmens bzw. eines Bereichs im Unternehmen, ganz gleich, ob es sich um ein zum Verkauf bestimmtes Standardprodukt oder um ein internes Produkt zur Effizienzsteigerung handelt.

**Produktvision (Product Vision)** Eine kurze Aussage über den gewünschten zukünftigen Zustand, der erreicht werden würde, wenn man ein Produkt entwickelt und ausliefert. Eine gute Vision sollte einfach und klar formuliert sein und den Personen, die sie umsetzen sollen, eine klare Richtung weisen.

**Produktziel (Product Goal)** siehe Scrum Guide 2020

**Quality Gate** Quality Gates sind Punkte im Ablauf eines Entwicklungsprojekts, bei denen anhand von im Voraus eindeutig bestimmten Qualitätskriterien über die Freigabe des nächsten Projektschrittes entschieden wird. In Scrum dienen die „Definition of Ready“ (DoR) und die „Definition of Done“ (DoD) als Checklisten zur Überprüfung.

**Rahmenwerk (Framework)** Ein Rahmenwerk bildet einen Rahmen von Aspekten ab, die für die Erreichung eines bestimmten Ziels beachtet werden sollten. Im Gegensatz dazu gibt eine Norm eine Reihe von Aspekten vor, die zur Erreichung eines Ziels zwingend zu beachten sind.

**Randbedingungen/Anforderungen (Constraints/Requirements)** Anforderungen, die den Lösungsraum jenseits dessen einschränken, was notwendig ist, um die funktionalen und qualitativen Anforderungen zu erfüllen. Beispiel: Das Produkt muss die Vorschriften der DSGVO einhalten. Oder: Das Produkt muss auf Windows 10 lauffähig sein.

**Refinement Meeting** Vereinbarter Event zur Durchführung des Product Backlog Refinements.

**Release (Release)** Eine Produktversion, die ausgeliefert werden soll. In der Regel besteht ein Release aus mehreren umgesetzten Anforderungen, also aus mindestens einem oder mehreren Inkrementen. Dabei ist ein Inkrement nicht zwingend auch ein Release. Der Product Owner entscheidet dabei, ob ein Inkrement wertvoll genug ist, um ein Release zu sein (und somit an den Kunden ausgeliefert wird).

**Release-Plan (Release Plan)** Planung, welche Anforderungen in einem ausgelieferten (Teil-)Produkt (Release) umgesetzt werden sollen. Der Release-Plan basiert auf dem Produktplan. Basis für die Release-Planung ist einerseits das immer sortierte und geschätzte Product Backlog (alles, was wir noch zu erledigen haben) und andererseits die Entwicklungsteams und deren Kapazitäten.

**Release Train (Release Train)** Ein Ansatz, mit dem Visionen, Planung und gegenseitige Abhängigkeiten vieler Entwicklungsteams aneinander ausgerichtet werden, indem auf der Grundlage einer gemeinsamen Kadenz (identische Sprint-Länge) eine teamübergreifende Synchronisierung ermöglicht wird. Ein Release Train konzentriert sich auf einen schnellen, flexiblen Flow auf der Ebene eines größeren Produkts.

**Retrospective** siehe Sprint Retrospective

**Return on Investment (ROI)** Allgemein: eine betriebswirtschaftliche Kennzahl zur Messung der Rendite einer unternehmerischen Tätigkeit, gemessen am Erfolg im Verhältnis zum eingesetzten Kapital.

Hier: Methode zur Priorisierung der Anforderungen an ein Produkt im Product Backlog.

**SAFe (SAFe)** siehe Scaled Agile Framework (SAFe)

**Scaled Agile Framework (SAFe)** Framework zur Skalierung von agilem Arbeiten in großen Umgebungen.

**Schätzarten (Kinds of Estimation)** Unterschiedliche Arten zu schätzen. Siehe dazu z. B.: Planning Poker, Affinity Estimation, Magic Estimation.

**Schätzeinheit (Estimation Unit)** Eine Schätzeinheit definiert, nach welchen Maßen Schätzungen durchgeführt werden. Übliche Schätzeinheiten sind Zeit (in Minuten, Stunden, Tagen, ...), Entfernung (in Metern, Meilen ...). In Scrum werden der Arbeitsaufwand und/oder die Komplexität eines Arbeitspaketes (User Story) sehr gerne in relativen Größen geschätzt (Arbeitspaket A ist fünfmal umfangreicher als Paket B), da in der Regel die benötigte Zeitspanne zur Umsetzung einer Arbeit nur schwer eingeschätzt werden kann. Als Wertevorrat für die Schätzungen kommen beispielsweise Kleidergrößen (XXS bis XXL) oder Zahlenreihen wie beispielsweise eine abgewandelte Fibonacci-Folge (0, 0.5, 1, 2, 3, 5, 8, 13, 20, 40, 100) infrage. Wichtig ist, dass der verwendete Wertebereich eine ausreichende Spreizung zwischen den einzelnen Werten aufweist, damit der Schätzfehler „automatisch“ berücksichtigt wird. – Schätzen bedeutet nicht Wissen, sondern nur „es könnte so sein“.

**Schätzkonzept (Estimation Concept)** Ein Schätzkonzept ist eine Planung, wie in einem Team oder Projekt geschätzt werden soll. Dies beinhaltet sowohl den Schätzprozess als auch die Schätzarten sowie die Schätzeinheiten. Üblicherweise wird ein Schätzkonzept zu Beginn einer Produktentwicklung / eines Projektes definiert und in einer Retrospektive an die neuen Erkenntnisse angepasst.

**Schätzung (Estimation)** Eine Schätzung ist eine grobe Berechnung von ungefähren Werten, einer ungefähren Anzahl/Menge oder einem ungefähren Ausmaß eines Vorhabens. In Scrum werden meist Backlog-Elemente in ihrem erwarteten Aufwand geschätzt, um die Planung für einen Sprint durchführen zu können und zu wissen, wie viele Elemente ungefähr ins nächste Sprint Backlog passen.

**Scrum Master (Scrum Master)** siehe Scrum Guide 2020

**Scrum of Scrums (Scrum of Scrums)** Ein Ansatz zum Koordinieren der Arbeit der Developer, die zur Umsetzung der Arbeiten in einzelne Spezialistenteams aufgeteilt wurden. Zur Koordination der Teams untereinander treffen sich Vertreter der einzelnen Teams nach Erfordernis mehrmals im Sprint. Der Ablauf eines Scrum of Scrums ähnelt dem Daily Scrum.

**Scrum-Planungsprinzipien (Scrum Planning Principles)** Alle Planungen in Scrum basieren auf folgenden Planungsprinzipien:

1. Pläne können nicht im Voraus richtig sein: Statt alles am Anfang zu planen (→ Unsicherheiten, die Zukunft kann andere Fakten/Anforderungen/Erkenntnisse bringen), folgt Scrum seinen empirischen Wurzeln, die in den Prinzipien von Untersuchung (Inspection) und Anpassung (Adaptation) liegen.
2. Vorabplanung sollte hilfreich, aber nicht exzessiv sein: Es reicht, wenn die wesentlichen Eckpunkte zu Beginn feststehen. Die Kunst liegt darin, die richtige Balance zwischen den im Vorfeld getroffenen Vorhersagen (Entscheidungen) und den Just-in-Time-Anpassungen zu finden.
3. Planungsbestand richtig organisieren: Zu viel Planung im Voraus birgt in adaptiven Projekten die große Gefahr von Fehlentscheidungen, die später wieder korrigiert werden müssen.
4. Planungsoptionen so lange wie möglich/nötig offenhalten: Entscheidungen sollten am besten dann getroffen werden, wenn alle Informationen bekannt sind und damit das Risiko einer Fehlentscheidung am geringsten ist.
5. Schnelles Lernen und Abweichen vom Plan, wenn nötig: Vorab Geplantes muss an neue Erkenntnisse angepasst werden.

6. Konzentration auf Anpassen statt auf Befolgen eines (festen) Plans: „Planung ersetzt Zufall durch Irrtum“ – Irrtümer sollten so schnell und so nachhaltig wie möglich korrigiert werden.
7. Häufigere kleinere Releases reduzieren das Risiko: Kleine Releases benötigen weniger Entwicklungszeit und -ressourcen und tragen schneller zu einem positiven Cashflow bei. Außerdem ermöglichen sie ein schnelleres Feedback vom Kunden und erlauben ein schnelles Reagieren auf Fehlentwicklungen und/oder Anforderungsänderungen.

**Skalieren von Scrum (Scaling Scrum)** Mehrere Teams arbeiten zusammen an der Erstellung eines Produktes. Dafür haben sich mehrere Frameworks etabliert, wie beispielsweise Nexus, Scaled Agile Framework (SAFe), LeSS, Scrum@Scale.

**Sprint** siehe Scrum Guide 2020

**Sprint Backlog** siehe Scrum Guide 2020

**Sprint-Burn-down-Chart** Grafische Darstellung, wie viel vom geplanten Aufwand (z. B. gemessen in Story Points) im aktuellen Sprint noch abzuarbeiten ist. Die Summe der Story Points aller in der Sprint-Planung für diesen Sprint ausgewählten User Stories ergibt das Soll der im Sprint abzuarbeitenden Story Points. Damit das Team leicht erkennen kann, wie es mit seiner Arbeit im Plan liegt, trägt man in die Grafik eine Ideallinie (Diagonale von links oben nach rechts unten) zur Abarbeitung aller Story Points ein. Die Reallinie ergibt sich durch Eintrag der täglich abgearbeiteten Story Points. Liegt die Reallinie oberhalb der Ideallinie, dann ist das Team im Zeitverzug. Andernfalls liegt es gut im Zeitplan.

**Sprint Planning** siehe Scrum Guide 2020

**Sprint Retrospective** siehe Scrum Guide 2020

**Sprint Review** siehe Scrum Guide 2020

**Sprint-Ziel (Sprint Goal)** siehe Scrum Guide 2020

**Stakeholder** Personen oder Organisationen, die ein Produkt / eine Dienstleistung beeinflussen (z. B. Auftraggeber, Gesetzgeber, Architekten, Entwickler) oder in irgendeiner Weise davon betroffen sind (z. B. Nutzer, Wartungspersonal).

**Stand-up Meeting** siehe Daily Scrum

**Story Mapping** Story Map(ing) dient der Visualisierung des Product Backlogs und kann auch als Roadmap verwendet werden. Eine Story Map liefert eine ganzheitliche Sicht auf das Product Backlog aus Sicht der Anwender und des abzubildenden Geschäftsprozesses. Sie steigert das Verständnis für die Ziele und die geforderten Funktionalitäten und deckt Lücken in der Anforderungsbeschreibung bzw. im Product Backlog auf. Eine gute Story Map hilft bei der Formulierung des Minimum Viable Product und der Release-Planung.

**Story Points** Story Points sind eine Schätzeinheit, nach der Schätzungen durchgeführt werden. Dabei sind Story Points eine relative Einheit, die immer in Bezug auf ein oder mehrere Referenzelemente gestellt werden. Meist wird als Story Points dabei eine abgewandelte Fibonacci-Reihe verwendet (0, 0.5, 1, 2, 3, 5, 8, 13, 20, 40, 100).

**Task Board** Eine Auflistung, die von dem Umsetzungsteam (Entwickler) erstellt und verwendet wird, um die Aufgaben, die aus den Sprint-Backlog-Einträgen (Sprint Backlog Items) entstehen, zu verwalten.

**Technische Schulden (Technical Debt)** Technische Schulden sind bewusste Ungenauigkeiten oder bekannte Fehler, die akzeptiert werden, um ein Inkrement schneller „abzuschließen“. Diese technischen Schulden verursachen in späterer Folge hohe Nacharbeiten, was nachfolgende Inkremente verzögert.

**Test** Tests sind methodische Versuche, mit denen festgestellt werden soll, ob Eigenschaften oder Leistung einer Sache, einer Person oder einer Hypothese den Erwartungen entsprechen. Im Rahmen von Scrum werden die Inkremente darauf hin getestet, ob sie den an sie gestellten Anforderungen entsprechen.

**Test-Driven Development (TDD)** Die eigentliche Programmierung erfolgt in kleinen, wiederholten Mikroiterationen. Eine solche Iteration, die nur wenige Minuten dauern sollte, hat drei Hauptteile, die man im englischen Red-Green-Refactor nennt:

1. Red: Schreibe einen Test.
2. Green: Ändere den Programmcode mit möglichst wenig Aufwand ab und ergänze ihn, bis er nach dem anschließend angestoßenen Testdurchlauf alle Tests besteht.
3. Refactoring: Räume dann im Code auf.

**Thema (Theme)** Ein Thema ist die (logische) Zusammenfassung mehrerer Bereiche im Product Backlog. In der Regel besteht ein Thema aus mehreren Epics und/oder User Stories oder Features.

**Time2Market (Time2Market, TTM, T2M)** oder Time-to-Market (TTM) ist die Zeit, die ein Produkt von der Konzeption bis zum Verkauf benötigt. Der Grund, warum die Markteinführungszeit so wichtig ist, liegt darin, dass sich eine verspätete Produkteinführung in jeder Branche negativ auf den Produkterfolg auswirkt.

**Timebox (Time Box)** Eine Timebox ist ein fester Zeitrahmen für einen Vorgang/Event. Ein Timebox-Event endet immer, wenn die dafür vorgesehene Zeit abgelaufen ist, unabhängig davon, ob alle Arbeiten auch abgeschlossen werden konnten.

**Triangulation** Die Triangulation ist eine Schätzart, bei der ein zu schätzendes Element immer in Verbindung zu zwei anderen Elementen in seinem Aufwand geschätzt wird. Dabei werden ein vermutlich kleineres Element und ein vermutlich größeres Element als das zu schätzende Element als Referenz hergenommen. Das zu schätzende Element wird dann in Relation zu den zwei Referenzelementen gestellt.

**True Leader** Ein True Leader (also eine wahre Führungspersönlichkeit) unterstützt die Menschen dabei, selbstverantwortlich zu handeln und dafür auch einzustehen. Hierzu unterstützt ein True Leader die Menschen, indem er sie bei ihrer Selbstreflexion unterstützt, um neue Potenziale erkennen und nutzbar machen zu können.

**Überprüfung (Inspect)** siehe Scrum Guide 2020

**User Story** Eine User Story beschreibt die Anforderungen aus Sicht der Anforderer in einer verständlichen Sprache. Sie dient dem gemeinsamen Verständnis und sollte als Diskussionsgrundlage gewählt werden. Eine User Story wird anhand folgender Satzschablone gebildet: Als <Rolle> möchte ich <Ziel>, damit ich <Nutzen/Begründung>.

**Velocity** Die Velocity ist die Leistungsfähigkeit eines Teams im Zuge einer Iteration. Berechnet wird sie durch die durchschnittlich ABGESCHLOSSENE (laut DoD) Arbeit der bisherigen Iterationen.

**WIP-Limit** Durch geeignete WIP-Limits (→ Work in Progress) lassen sich die Durchlaufzeiten, also die Zeit zur Fertigstellung von Arbeitsergebnissen, verbessern. In Scrum bestimmt das Entwicklungsteam das WIP-Limit.

**Work in Progress (WIP)** Summe aller Arbeitspakete, die in Bearbeitung sind.