

# AGILE HORIZN

NEXT LEVEL EDUCATION

MASTERCLASS  
SCRUM 150 INKL.  
POWERBOOK  
PRÜFUNGS-  
VORBEREITUNG  
SCRUM.ORG | PSM1

# INHALT

---

## 1. DEIN WEG ZUR ZERTIFIZIERUNG

## 2. SCRUM IM ÜBERBLICK

## 3. SCRUM ROLLEN

## 3. - 4. EVENTS

## 4. ARTEFAKTE

## 5. - 7. DEFINITION UND WERTE

## 8. FÜR DIE PRÜFUNG

### **Hinweis:**

Die MasterClass | "Scrum nach 150 Minuten anwenden", hat dir das perfekte Rüstzeug für den Alltag mitgegeben.  
In diesem Handout kannst du noch einmal alles nachlesen.

Falls du die Prüfung zum Scrum.org PSMI [Professional Scrum Master I] ablegen willst, beachte bitte unsere Lerntipps.

Die Sicherheit zum Bestehen, können wir dir natürlich nur einräumen, wenn du gesondert unsere Prüfungsvorbereitung besuchst.

In der Regel schaffst du aber mit einem Aufwand von 2-3 intensiven Lernwochen die Prüfung auch ohne gesondertes Training von uns.

# DEIN WEG ZUR OPTIONALEN ZERTIFIZIERUNG

**DAMIT DU DIE PRÜFUNG BEIM ERSTEN MAL BESTEHST, EMPFEHLEN WIR DIR  
DIE FOLGENDEN SCHRITTE:**

- Die **Prüfungssprache ist English** – lerne also auch am Besten in English
- **ScrumGuide** (Nachschlagewerk der beiden Scrum Erfinder Ken Schwaber und Jeff Sutherland)  
lesen und markieren - Anhang zu diesem Powerbook
- **Agile Manifesto** lesen und markieren - Seite 5 in diesem Powerbook
- Die **12 agilen Prinzipien** lesen und markieren - Seite 6 -7 in diesem Powerbook
- **Scrum.org Prüfungsinfos & Themen** beachten:  
<https://www.scrum.org/assessments/professional-scrum-master-i-certification>
- **Scrum Open Assessment** (<https://www.scrum.org/open-assessments/scrum-open>) | so lange  
üben bis Du mind. 95% der Fragen richtig beantwortest. Wichtig: Das Open Assessment ist  
deutlich einfacher, als der echte Test!
- **PSM I Exam Simulator** - Simulator für 22€ kaufen: <https://mplaza.training/online-courses/scrum-framework/> | so oft wie möglich nutzen, bis du dich 100% sicher fühlst.
- **Zertifizierungs-Zugang** auf Scrum.org kaufen:  
<https://www.scrum.org/assessments/professional-scrum-master-i-certification>

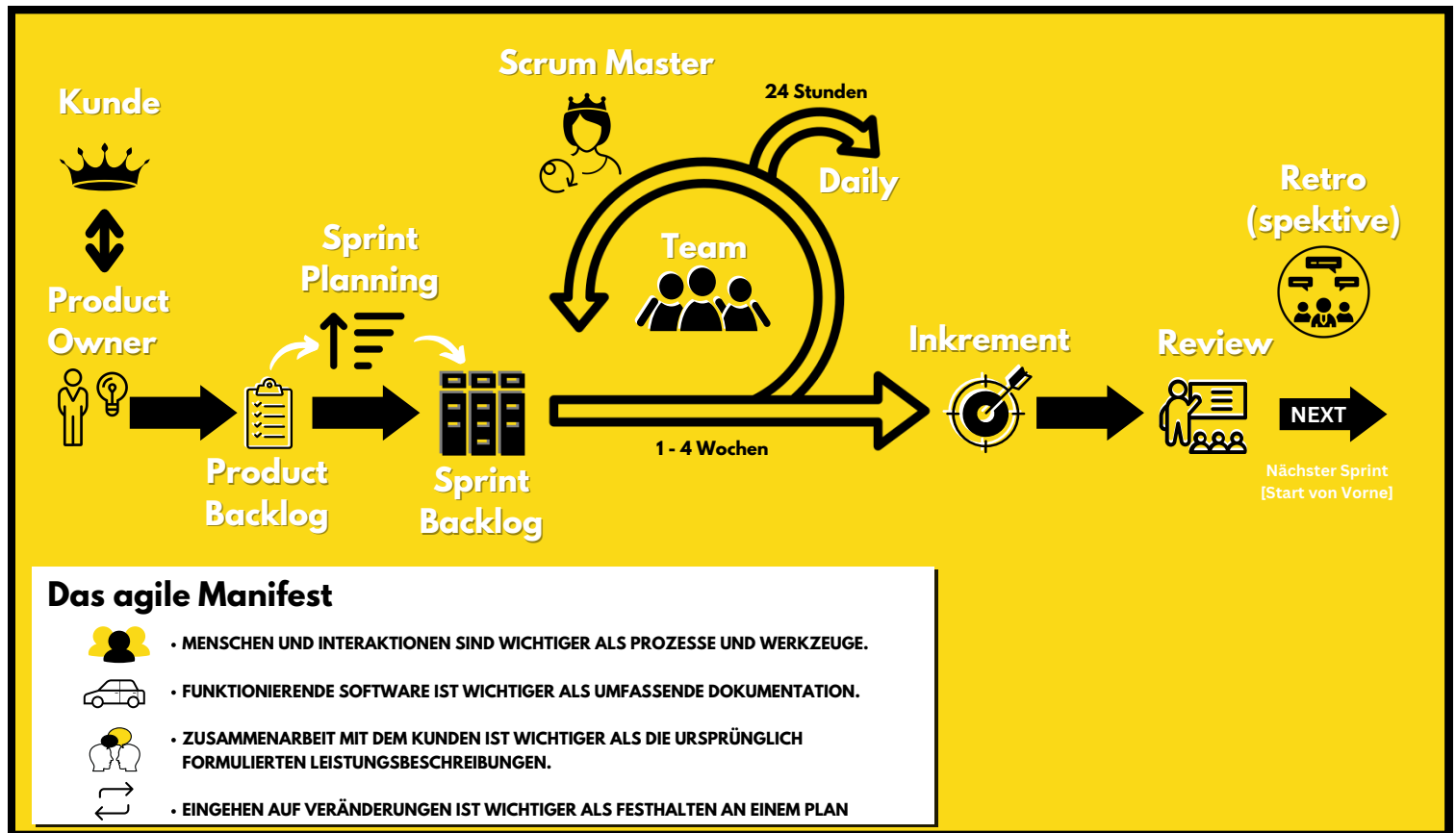
**FRÜHESTENS NACH 2 INTENSIVEN LERN-WOCHEN [PARETTO: 80/20 PRINZIP KLAPPT NICHT!]**

**Zertifiziere Dich erfolgreich auf scrum.org**

(<https://www.scrum.org/professional-scrum-master-i-certification>)

- 80 Multiple Choice Fragen
- 60 Minuten Zeit. // Mindestens 85% der Fragen müssen korrekt sein! Das Pareto Prinzip: 80/20 klappt also nicht!
- Übersetzer bereithalten – wir empfehlen [www.deepl.com](https://www.deepl.com) [oder Google Plug In -> empfehlen wir nicht!]
- Zeit im Auge halten und „Bookmarks“ bei schwierigen Fragen setzen  
[ca. 10 Min. vor Ende Bookmarks prüfen]
- Empfehlung spätestens 1-2 Wochen nach dem Training den Test machen!

# SCRUM IM ÜBERBLICK



## Sprint

max. 4 Wochen lang

**Planning**

max. 8 Std.  
[bei einem 4 Wochen Sprint]

**Daily Scrum**

max. 15 min.

**Review**

max. 4 Std.  
[bei einem 4 Wochen Sprint]

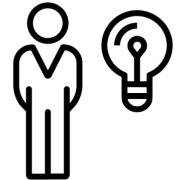
**Retro-spektive**

max. 3 Std.  
[bei einem 4 Wochen Sprint]

# SCRUM ROLLEN

## Product Owner

- Auftraggeber
- pflegt & priorisiert Backlog
- vertritt Interessen des Kunden / Stakeholders
- passiv im Daily Scrum
- steht für Rückfragen zur Verfügung
- ist nicht der Chef des Scrum Teams
- verändert nichts im Backlog während des Sprints



## Scrum Master

- trägt Verantwortung für Scrum Prozess
- ist Vermittler und Unterstützer
- sorgt für Informationsfluss zwischen Product Owner und Team
- moderiert Scrum-Meetings
- ist nicht der Chef des Scrum Teams



## Development-Team (3-9 Personen)

- multidisziplinär und selbststeuernd
- besteht aus drei bis neun Personen
- ist sein eigener Manager
- trifft sich täglich zum Daily Scrum
- lässt sich nicht die Arbeitsweise vom Product Owner vorschreiben
- berichtet im Daily nicht an Scrum Master / PO → Team Member berichten einander



# EVENTS

## Sprint [max. 4 Wochen]

- Iterativ
- Dauer von 1 bis maximal 4 Wochen
- Team bearbeitet ausgewählte Menge an Aufgaben

## Sprint Planning [max. 6h - 2h je Sprint-Woche]

- Beantwortet zwei Fragen:
  - 1) Was kann im kommenden Sprint entwickelt werden?
  - 2) Wie wird die Arbeit im kommenden Sprint erledigt?
- Jeder Sprint beginnt mit dem Planning
- Product Owner stellt die zu bearbeitenden Aufgaben vor
- Team wählt Items aus dem Backlog, die in
- einem Sprint bewältigt werden

**Daily Scrum** [max. 15 min.]

- Tägliches Meeting des Development Teams von 15 Minuten zwecks Informationsaustausch
- Beantwortung der drei Fragen und macht damit Probleme transparent:
  - 1) Was ist seit dem letzten Meeting geschehen?
  - 2) Was plane ich bis zum nächsten Meeting?
  - 3) Was hindert mich daran, dieses Ziel zu erreichen?

**Sprint Review** [max. 4h - 1h je Sprint-Woche]

- Ergebnisvorstellung des Inkrements am Ende jeden Sprints durch das Team
- Ziel ist Darstellung erreichter Fortschritte
- Einholung Stakeholder Feedback

**Sprint Retrospektive** [max. 3h - 45 min je Sprint-Woche]

- Sprint wird mit Retrospektive abgeschlossen
- Team blickt auf Arbeitsprozess des letzten Sprints zurück
- Ergebnis einer Retrospektive: a) Beibehalten von Gutem b) Veränderung von Schlechtem
- Emotionen erlaubt und erwünscht

# ARTEFAKTE

---

**Backlog**

- Product Backlog enthält alle Anforderungen an das Produkt formuliert in sog. User Stories
- Wird im Laufe der Projektlaufzeit vom Product Owner weiterentwickelt und gepflegt
- Aus dem Product Backlog zieht sich das Team die User Stories für den nächsten Sprint in das Sprint Backlog
- Die User Stories im Sprint Backlog werden im Sprint Planning nach Aufwand geschätzt und ggfs. weiter ausdefiniert (Product Owner ist anwesend)

**Definition of Done**

- Legt fest welche Anforderungen erfüllt sein müssen, damit eine Anforderung bzw. User Story abgeschlossen werden kann (siehe auch Akzeptanzkriterien)
- Wird gemeinsam vom Team und Product Owner festgelegt

**Increment**

- Die Summe aller nach einem Sprint gemäß Definition of Done fertiggestellten Anforderungen (bzw. User Stories) sowie der bereits zuvor fertiggestellten (aktueller Stand des Produktes)
- Das Increment muss „done“, also potentiell nutzbar und veröffentlichbar („releasable“) sein

# DEFINITION & WERTE

**Scrum (Deutsch: Gedränge; Begriff aus dem Rugby) ist ein Vorgehensmodell** aus der agilen Softwareentwicklung, bei dem Projekte iterativ, inkrementell und in sogenannten Sprints sehr schnell umgesetzt werden.

Scrum basiert auf der Annahme, dass komplexe Projekte nicht detailliert planbar sind und daher in kleinere Aufgaben heruntergebrochen und Stück für Stück bearbeitet werden müssen, während sich die Planbarkeit nach und nach erhöht.

**Da Projektgeschäft mit Lernen, Innovation und Überraschungen einhergeht, werden bei Scrum kurze Entwicklungsschritte (Sprints) in schneller Abfolge mit möglichst wenigen Abhängigkeiten bearbeitet.**

So können sowohl das resultierende Produkt als auch die Wirksamkeit der aktuellen Praktiken überprüft und anschließend die Projektziele und Prozesspraktiken immer wieder angepasst werden (Inspect & Adapt). Damit stellt Scrum ein anpassbares Framework dar, das auf definierten Prinzipien, Werten, Rollen, Artefakten und Ereignissen basiert.

## Scrum basiert auf den Prinzipien:

- Transparenz
- Überprüfbarkeit
- Anpassung

2001 wurden die Werte der agilen Software-Entwicklung von Jeff Sutherland, Ken Schwaber und weiteren Experten im **Agilen Manifest** zusammengefasst:



**Individuen und Interaktionen** sind wichtiger als Prozesse und Tools



**Zusammenarbeit mit dem Kunden** ist wichtiger als die Vertragsverhandlung



**Funktionierende Software** ist wichtiger als eine umfassende Dokumentation



**Reagieren auf Veränderung** ist wichtiger als das Festhalten an einem Plan

- |                                |             |                             |
|--------------------------------|-------------|-----------------------------|
| • Individuals and interactions | <b>over</b> | processes and tools         |
| • Working software             | <b>over</b> | comprehensive documentation |
| • Customer collaboration       | <b>over</b> | contract negotiation        |
| • Responding to change         | <b>over</b> | following a plan            |

# DEFINITION & WERTE

## Die 12 agilen Prinzipien:

Diese machen deutlich, dass die Anfänge des Agilen Projektmanagements geprägt waren von einer Ablehnung des traditionellen Projektmanagements mit intensiver Planung, Überwachung und Steuerung.

- Unsere höchste Priorität ist es, Kundinnen und Kunden durch **frühe und kontinuierliche Auslieferung wertvoller Software (Produkte)** zufrieden zu stellen.
- Radikale Anforderungsänderungen sind selbst spät in der Entwicklung willkommen. **Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil der Kundinnen und Kunden**
- **Liefere funktionierende Software (Produkte) regelmäßig innerhalb weniger Wochen** oder Monate und bevorzuge dabei die kürzere Zeitspanne.
- Fachkundige und Entwickelnde müssen während des Projektes **täglich zusammenarbeiten.**
- Errichte Projekte rund um **motivierte Individuen.** Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
- Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, **ist im Gespräch von Angesicht zu Angesicht.**
- **Funktionierende Software** ist das wichtigste Fortschrittsmaß.
- Agile Prozesse fördern eine nachhaltige Entwicklung. Die Auftraggebenden, Entwickelnden und Benutzenden **sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.**
- Ständiges **Augenmerk auf technische Exzellenz und gutes Design** fördert Agilität.
- Einfachheit – **die Kunst, die Menge nicht getaner Arbeit zu maximieren** – ist essenziell.
- Die besten Architekturen, Anforderungen und Entwürfe entstehen durch **selbstorganisierte Teams.**
- **In regelmäßigen Abständen reflektiert das Team,** wie es effektiver werden kann und passt sein Verhalten entsprechend.



# DEFINITION & WERTE

## 12 agile principles

1

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4

Business people and developers must work together daily throughout the project.

5

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7

Working software is the primary measure of progress.

8

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9

Continuous attention to technical excellence and good design enhances agility.

10

Simplicity—the art of maximizing the amount of work not done—is essential.

11

The best architectures, requirements, and designs emerge from self-organizing teams.

12

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

# FÜR DIE OPTIONALE PRÜFUNG

8

## Das solltest du Dir merken:

- Scrum ist ein Framework – nicht mehr aber auch nicht weniger.
- Die wenigen "Regeln" sind alle einzuhalten.
- Scrum wird in unsicheren / komplexen Umfeldern eingesetzt.
- Durch Scrum sollen Kreativität, Innovation und Produktivität gefördert werden.
- Das Scrum Team besteht aus Product Owner, Scrum Master und dem Development Team.
- Der Product Owner ist ein value optimizer.
- Die Definition of Done ist eine Guideline für das gesamte Scrum Team, das Development Team hat den Hut dafür auf.
- Das Sprint Goal wird vom gesamten Scrum Team gemeinsam gesetzt, es zwischen Product Owner und Development Team verhandelbar

## Womit man immer richtig liegt:

Wenn man Agilität und Scrum im Kern verstanden hat, liegen die Antworten meist klar auf der Hand. So sind manche Dinge immer zutreffend, da sie die agile Denkweise widerspiegeln:

- Foster *collaboration* and *self-organization*
- Control and *reduce risk* using Scrum
- Give *forecasts*
- Rely on *empiricism*
- Be *transparent* – make information accessible for everyone
- *Improve* constantly
- *Done* = potentially shippable / releasable (good quality)
- *Change* is always welcome
- Every Sprint should produce a *done increment*

## Finger weg von diesen Antwortmöglichkeiten

- Es gibt keine ProjectManagerRolle in Scrum
- Scrum ist keine Methode (Methodology)
- Im Scrum Guide nicht definierte Begriffe – wie z.B ReleaseSprints– gibt es nicht
- Prozessualer Overhead wie (formally approved) lists/documents oder acceptance reports gibt es in Scrum nicht