



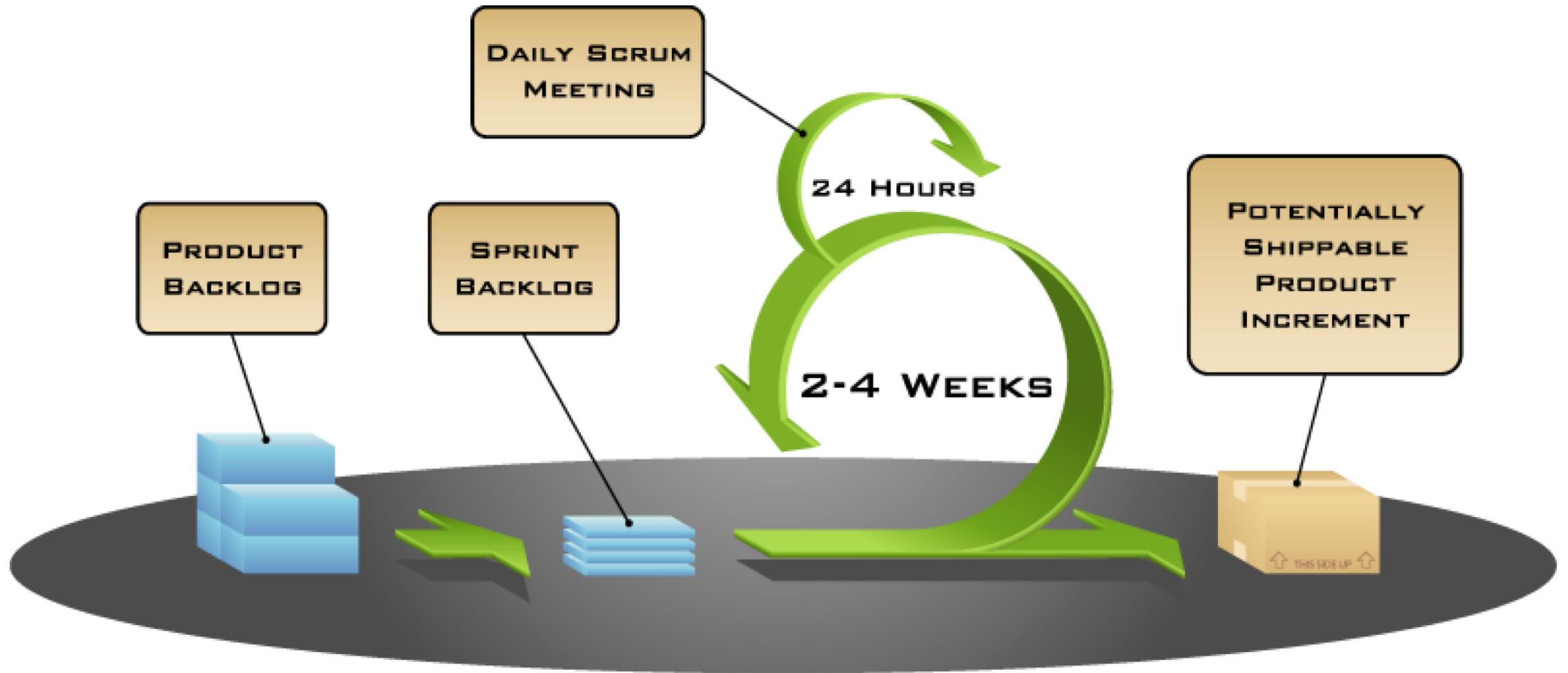
SCRUM-Primer

Agile Softwareentwicklung mit Scrum

Patrick Schäfer

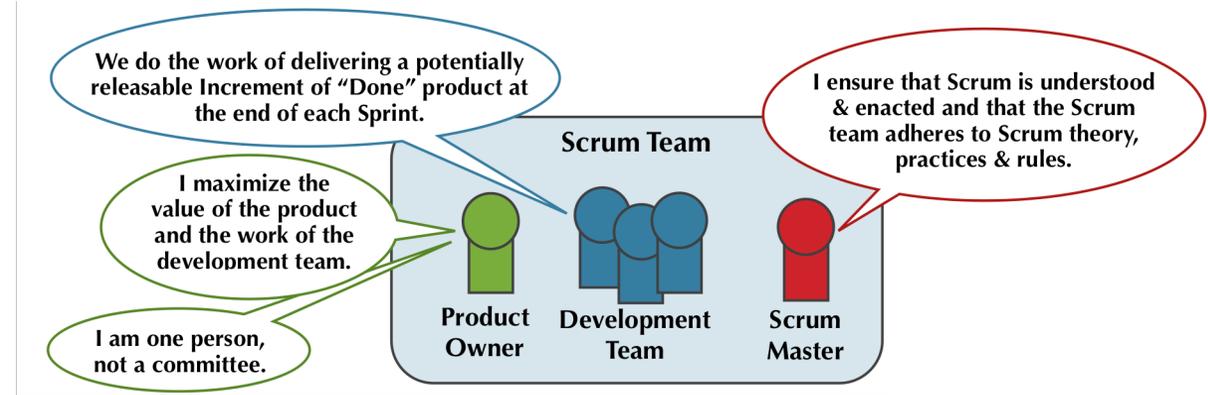
patrick.schaefer@hu-berlin.de

Semesterprojekt: Implementierung eines Brettspiels, WS 18/19



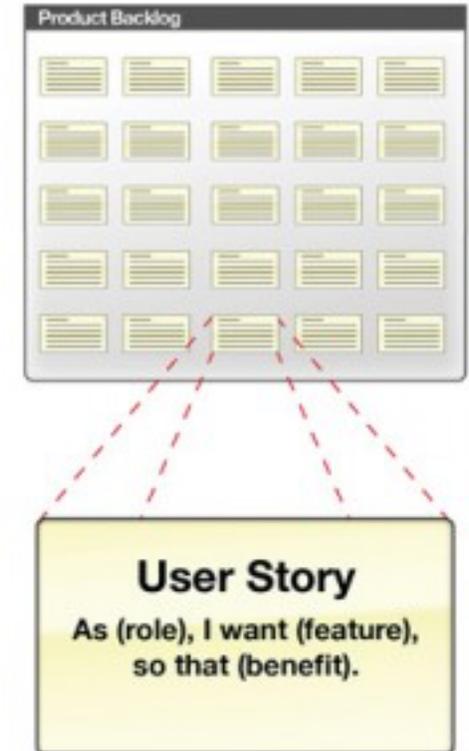
Rollen

- Product Owner (WIR):
 - Definition von Produkt-Features (User Stories)
 - Priorisieren der Features für die nächsten Sprints
- Scrum Master:
 - Beseitigt Hindernisse
 - Koordiniert das Team und die Meetings
 - Überwacht Projektfortschritt
- Entwicklungsteam (IHR):
 - Festes Team, jeweils 5 Personen
 - Entwickelt das Produkt eigenständig entsprechend der Liste der Features
 - Selbst-organisiert, hoher Grad an Eigenverantwortung
 - Entscheidet wie viele Features realisiert werden im Sprint



Product Backlog

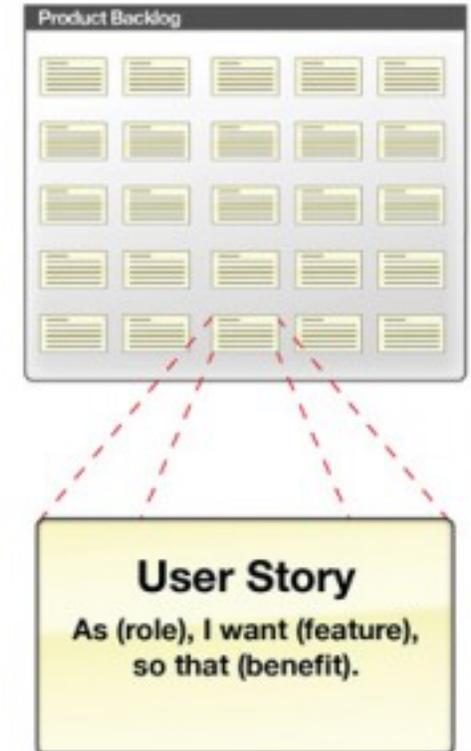
- Priorisierte Roadmap
 - Kunden-Features, Bugs, Technische Verbesserungen, ...
- Formuliert als „User Stories“
- Team schätzt Komplexität jeder User Story
- Wächst dynamisch:
 - Scrum Master darf Aufgaben hinzufügen
 - ABER: Priorisierung nur durch PO
- Hoher Detailgrad für User Stories mit hoher Priorität.



User Stories

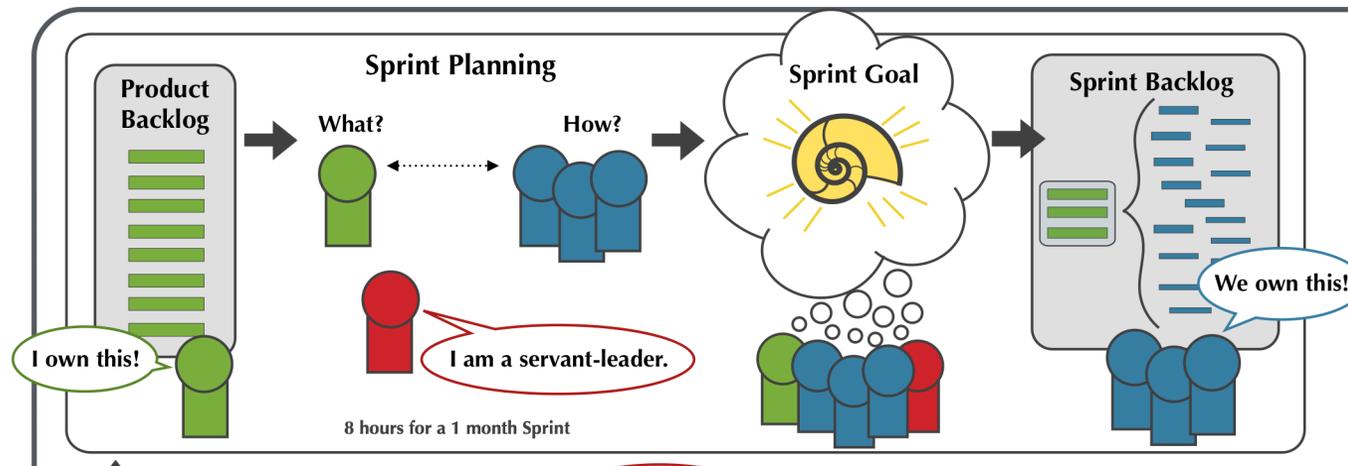
- As a ...
- I want ...
- So that I can ...

- Stories sollten klein genug sein, um sie in 1-2 Tagen fertigstellen zu können.
- Entwicklungsteam zerlegt komplexe Stories im kleinere Stories.



Sprint Backlog (pro Team)

- Teilmenge des Product Backlog an dem im aktuellen Sprint gearbeitet wird
- Komplexität jeder Story wird durch Entwicklerteam abgeschätzt
- Team entscheidet, wie viele Tasks in einem Sprint abgearbeitet werden
- **Aber:** Product Owner entscheidet in welcher Reihenfolge die Tasks bearbeitet werden



Sprint

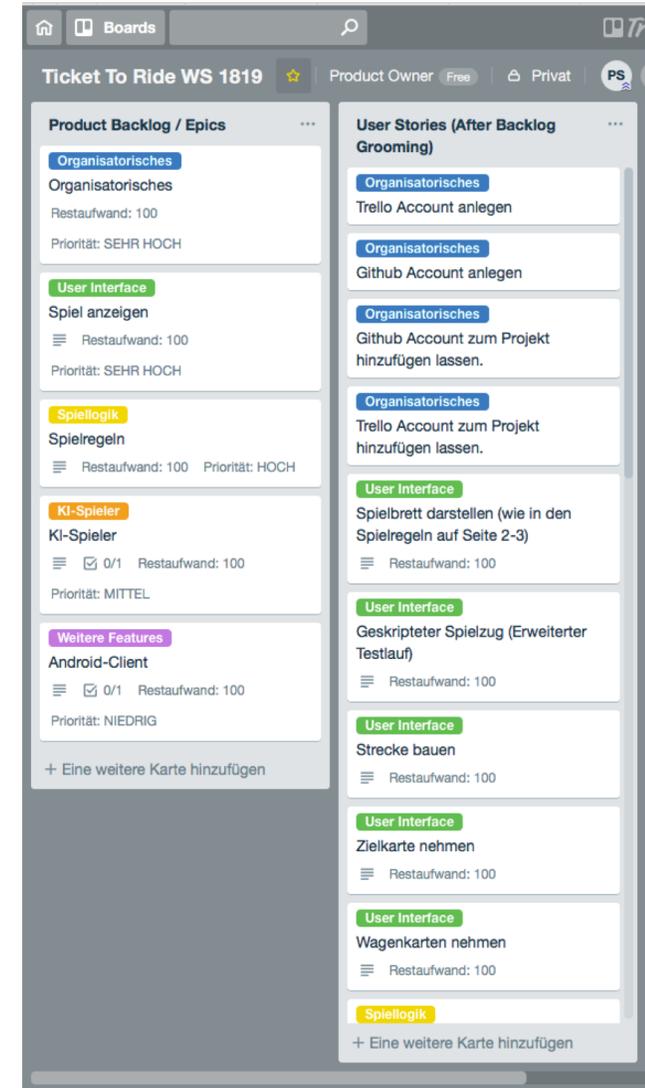
- Regelmäßiger kurzer Zyklus:
 - 2 Wochen, startet jeden 2ten Montag
- Fixer Zeitraum und feste Arbeitspakete (keine Änderungen)
- **Ziel: auslieferbares, getestetes Produkt! Wird am Ende des Sprints präsentiert**

Sprint Zeremonien

- Vorbereitend:
 - Backlog Grooming
 - Technical Refinement
- Sprint Planning
- Daily Scrum / Stand-up
 - 15 Minuten (idealerweise) jeden Tag
 - bei uns mindestens 2x pro Woche!
- Sprint Review / Demo
 - Das Team zeigt erreichten Projekt-Fortschritt im Sprint

Backlog Grooming des Product Backlog

- Montags alle 2 Wochen
- Vorbereitend zum nächsten Sprint
- Backlog Grooming:
 - PO erklärt Epics/User Stories im Product Backlog
 - PO und Team zerlegen Epics in User Stories
 - PO und Team überarbeiten User Stories
 - Definition von testbaren Akzeptanzkriterien



Technical Refinement

- 2-4h alle 2 Wochen
- Team zerlegt User Stories des Product Backlogs in technische Tasks
- Team schätzt Aufwand der Tasks

The screenshot displays a Jira board for the project "Ticket To Ride Team Gelb". The board is organized into several columns representing different stages of the development process:

- Development:** Contains five tasks, each with an effort estimate and an assignee. Tasks include: "Strecken-Priorität Strecke ist Nahe an Gegenstrecke" (S (3), MH), "Wagenkarten-Priorität Wagenkarten für Strecken mit hoher Priorität" (S (3), SH), "Optionen-Priorität in letzter Runde: Sinnvollen Zug auswählen (Strecke oder Bahnhof, keine Wagenkarten oder Zielkarten)" (M (8), BJ), "Wagenkarten-Priorität (im Nachhinein) für Farben senken/für verdeckt steigern" (S (3), SH), and "Informationen über die verbleibenden/abgeworfenen Wagenkarten -> Wahrscheinlichkeiten" (S (3), SH).
- (Automatische) Unit-Tests:** Contains five tasks: "Nach dem ersten mal ziehen darf beim zweiten mal auch verdeckt gezogen werden." (MH), "Parametrisierbare Regelbasierte KI (Interface)" (M (8), MH), "Strecken-Priorität nach Länge" (XS (1), SP), "Strecken-Priorität Tunnelmalus" (XS (1), SH), and "Strecken-Priorität nach Regelkonformität (genug Wagenkarten, ist frei)" (XS (1), BJ).
- Code Review:** Contains five tasks: "Testumgebung um die Leistung der KI's vergleichen zu können" (M (8), MH), "Repräsentation der Gegner" (S (3), SP), "Ermittlung einer guten Verbindungsstrecke für n Zielkarten" (L (20), MW), "alternative Streckenauswahl ohne längste Strecke (O)" (M (8), SH), and "Parametrisierung der Streckenauswahl (O)" (S (3), SH).
- Done:** Contains five tasks: "Priorisierung der Strecken: Wo muss dringend gebaut werden" (M (8), SH), "Repräsentation des Spielfelds" (M (8), SH), "Im laufe des Spiels Zielkarten ziehen" (Restaufwand: 100 Aufwand: M (8)), "Bahnhöfe bauen" (Restaufwand: 100 Aufwand: S (3)), and "Zugentscheidung" (Restaufwand: 100 Aufwand: M (8)).
- Done (vom PO abgenommen):** Contains five tasks: "KI Zusammenbasteln nach flow Chart." (S (3), BJ), "Einfaches Hauptmenü, welches einen Launcher mit den Werten initialisiert." (S (3), BJ), "Animationen: Unity einarbeiten" (S (3), SP), "Animation: Zielkarten ziehen" (M (8), SP), and "Animation: Karten Bezahlen" (S (3), SP).

Each task card also includes a "KI-Spieler" label, a "Restaufwand" (Remaining Work) indicator, and a "+ Eine weitere Karte hinzufügen" (Add another card) button at the bottom.

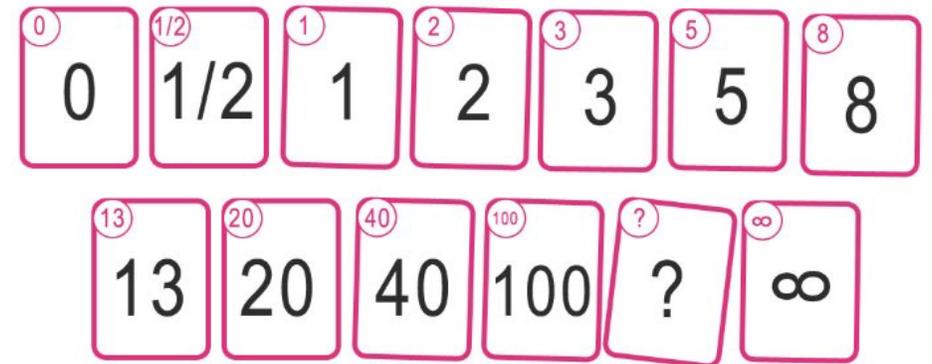
Sprint Planning

- Startet jeden Sprint: Auswahl von technischen Tasks
- Teil 1: Was (Product Owner, Product Backlog)
 - [Backlog Grooming: Klären der Einträge im Product Backlog]
 - Definition des Sprint Ziels
- Teil 2: Wie (Team, Sprint Backlog)
 - [Technical Refinement: Team zerlegt Stories]
 - [Technical Refinement: Team diskutiert (Komplexität) der Stories.]
 - Team definiert, was im Sprint geschafft wird, d.h., Tasks werden dem Sprint Backlog hinzugefügt

Aufwandsschätzung



T-Shirt-Größen



Story Points

Aufwandsschätzung

- Oftmals relative Komplexität / Größen
 - Story Points (Fibonacci Zahlen),
 - T-Shirt Größen,
 - Gummibärchen ...
- Baseline Story: kleinste Geschichte identifizieren.
- Stories in Relation zur Baseline abschätzen.

- Nur das Team schätzt

- Kapazität: Wie viel schafft das Team in einem Sprint:
 - $2 \times S + 1 \times M$ oder $2 \times M + 1 \times S$?

Planning Poker

- JEDES Teammitglied schätzt verdeckt Aufwand
- Schätzungen werden gleichzeitig aufgedeckt
- Weichen die *Schätzungen* weit:
 - Person mit höchster und niedrigster Schätzung legen Ihre Gründe für die Schätzung dar
 - Es wird wieder verdeckt geschätzt
- Die Komplexität ist kein Mittelwert sondern ein Schätzwert, hinter dem jede einzelne Person aus dem Team steht



Daily Scrum

- Kurzes Meeting (15 Minuten)
- Im Stehen (kurz)!
- Jedes Teammitglied beantwortet 3 Fragen:
 - Was wurde seit dem letzten Meeting erreicht?
 - Was wird bis zum nächsten Meeting erreicht?
 - Welche Hindernisse sind im Weg?
- Keine tiefgehenden Diskussionen!
- Diskussionen in anschließenden Meetings, falls notwendig
- Scrum Master hilft beim Lösen von Hindernissen



Fortschritt und Definition of Done

- Jedes Team aktualisiert die Schätzung des Restaufwands zur Fertigstellung seiner Arbeit im Spring Backlog: 100%, 50%, 10%, 0%
- „Done“:
 - Vom Team und Product Owner akzeptierte Kriterien
 - Ziel ist immer ein auslieferbares Produkt!
- Beispiel:
 - Implementiert
 - Peer-reviewed und Review abgenommen
 - Getestet: Unit Tests!
 - Product Owner hat die Story abgenommen

Sprint Review

- Am Ende eines Sprints
- Team präsentiert Ergebnisse
 - **DEMO der Software**, keine Power-Point Präsentation