

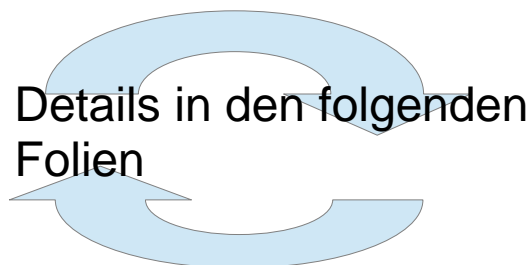
Scrum

Agile Entwicklung mit User Stories und Kanban

- **Product Owner** (Kunde bzw. Anforderer)
 - Kennt die Fachdomäne
 - Fachlicher Projektleiter
 - Erfolgsverantwortlich
- **Team der Softwareentwickler**
 - Umsetzung der User-Stories in Programme
 - Selbstverantwortliche (ohne Projektleiter)
- **Scrum-Master**
 - Verantwortlich für die Umsetzung der Scrum-Methode

Scrumteile

- **Vision**
- **Produktkonzept** (verbale Problembeschreibung)
- **Product Backlog** (Anforderungskatalog) mit User Stories (Sicht des Anwenders)
 - Alle User Storys
 - Priorisierung der UserStorys (MustHave, ShouldHave; Alternativ: MustHave, ShouldHave, CouldHaves, Wont have this time)
 - Aufwandsschätzung (Schwierigkeitspunkte – StoryPoints)
 - Ergänzung des Product Backlogs durch nicht-funktionale Anforderungen ohne User Stories z.B. Performance, Ausfallsicherheit usw.
- **Sprints**
- **Überprüfung**



Sprint

- **Sprint-Planung (1-4 Wochen), d.h. Umsetzung eines Teils der User Storys**
 - Auswahl der User Stories für die Umsetzung im Sprint
 - Ziele des Sprints
 - Kapazitätsfestlegung Sprint auf Grundlage der Schwierigkeitspunkte (Velocity=Geschwindigkeit) > Lernpotential
 - Analyse der User Stories
 - Design der User Stories durch Tasks (Funktionale oder objektorientierte Zerlegung der UserStories)
 - ---> Ergebnis: Sprint Backlog des ersten Springs

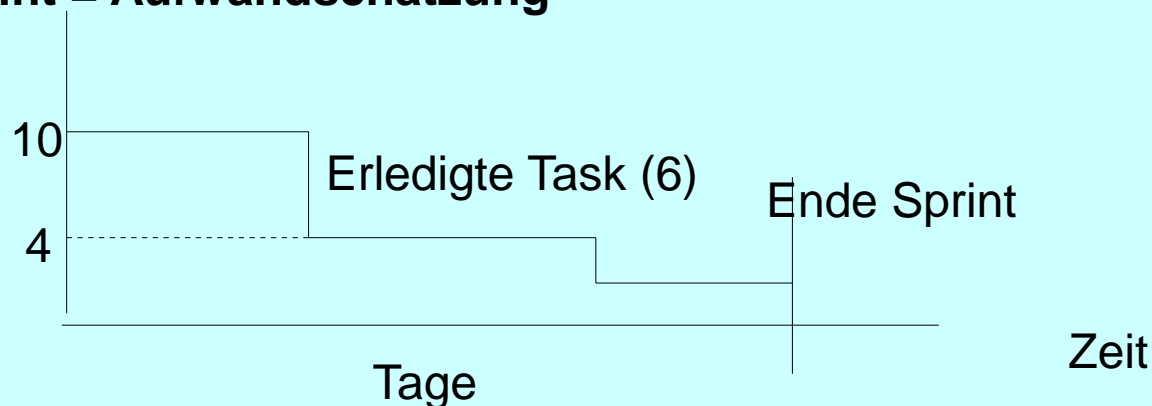


Sprint

• Sprint-Durchführung (Programmierung der User Stories)

- Beginn der Umsetzung
- Entwurf von Testszenarien (Sprint)
- Neue Erkenntnisse in Task umsetzen
- Tägliche Reflexion
 - Was habe ich gestern erreicht?
 - Was plane ich heute?
 - Welche Hindernisse oder Probleme haben sich ergeben?
- Aktualisierung des Scrum-Bords
- Aktualisierung des Sprint-Burndown-Carts

Story-Point = Aufwandschätzung



• Sprintende

- Überprüfung der Ergebnisse durch Kunde/E-Team
 - Öffentliche Vorführung der Sprint-Ergebnisse
 - Test der Sprints
 - Retrospektive mit Stärken und Schwächen des Entwicklungssprints
 - Ergebnis: Metaplantchnik mit Klassifizierung der Probleme und Stärken ---> Ableitung von Maßnahmen
- Maßnahmen: Verbesserung und Anpassung für nächsten Sprint

Stärken	Schwächen
Stärken/Schwächen	Abgeleitete Maßnahme

Releaseabschluss

- Integrationstest mit Testszenarien – Grundlage der Präsentation
- Endbeurteilung Entwicklungsprojekt - Stärken-Schwächen-Maßnahmen
- Dokumentation
- Beurteilung der Projektmitarbeiter

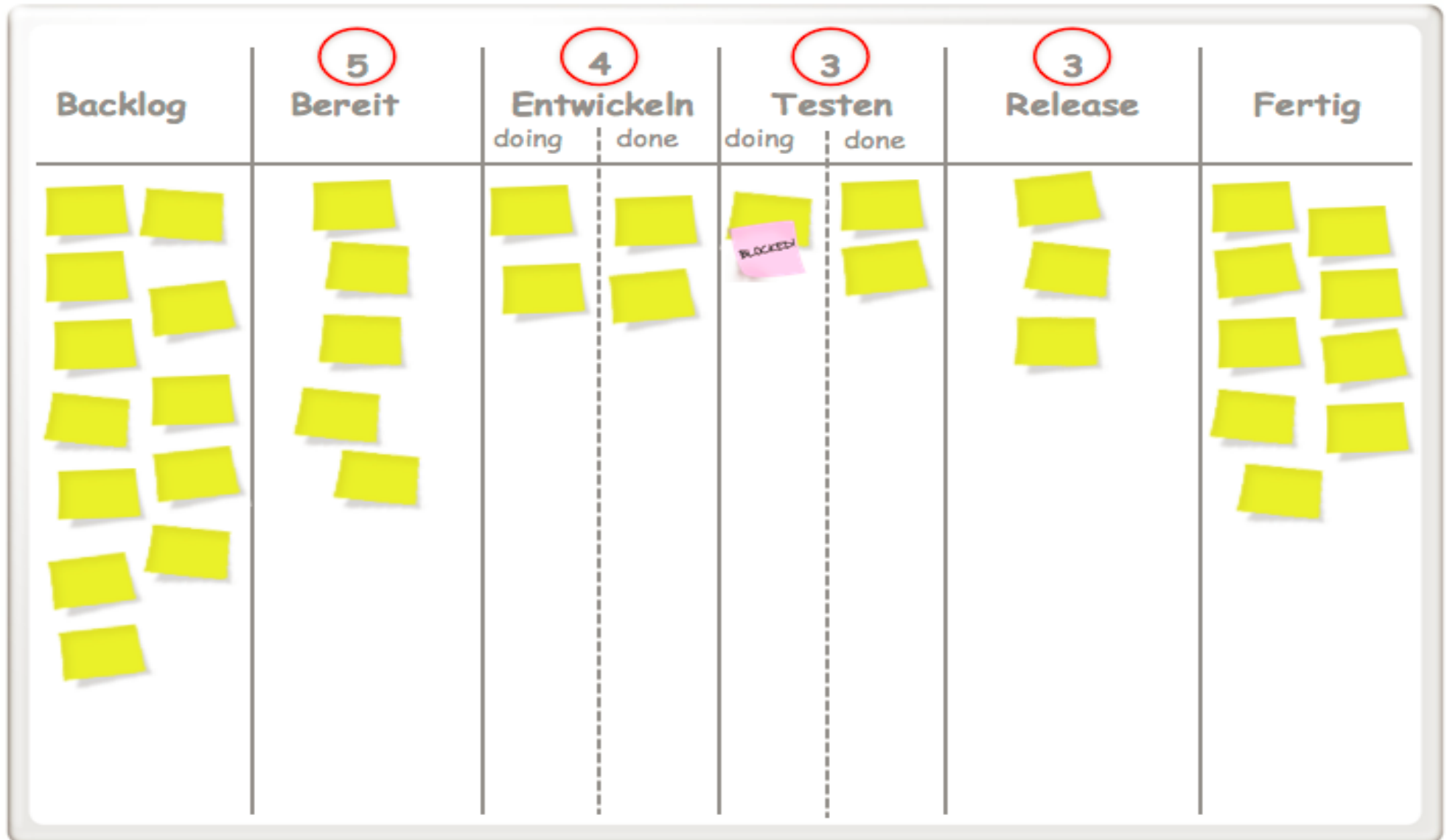
Kanban Eigenschaften

- Transparenz über Projektfortschritte
- Visualisierung von Problemen
- Reduziert die Durchlaufzeit
- Nutzungsbereiche: Entwicklung, Wartung, Ticketabarbeitung
- Geringe Widerstände bei der Einführung

Unterschiede Scrum - Kanban

Scrum	Kanban
Iterationen in gleichen Zeitabständen	Iteration kann frei festgelegt werden
Festlegung auf einen Arbeitsumfang	Arbeitsumfang kann festgelegt werden
Entwicklungsgeschwindigkeit des Teams beeinflusst die Arbeitsgeschwindigkeit	Durchlaufzeit wird als Erkenntnis weiterverwendet
Sprint definiert den Arbeitsumfang	Kein Sprint – kein definierter Arbeitsumfang
Schätzungen sind wichtiger Punkt	Schätzungen sind optional
Schreibt Rollen vor (PO, SM, T)	Keine Rolle
Team gehört das Scrum-Board	Kanbanboard für Team oder Einzelperson
Scrum-Board wird bei jedem Sprint neu erstellt	Kontinuierliche Kanban-Board-Verwendung
Priorisierung der Aufgaben im Backlog ist vorgeschrieben	Optionale Priorisierung

Kanban-Board



Noch Fragen?