Agile Softwareentwicklung
Scrum vs. Kanban

Betül Atliay, Ganna Shulika, Merve Yarat

Universität Salzburg

29. Jänner 2016

Inhalt

1. Agile Softwareentwicklung
2. Scrum
3. Kanban
4. Scrum vs. Kanban

ScrumMaster
Anforderungen
Taskboard
Agile
Scrum
Retrospektive
Backlog
Standup
Burn-down
Review
Kanban
Lean Beschätzung
Story Points
ProductOwner
Sprint
Software-Entwicklung
DoD
Projektmanagement

Agile Softwareentwicklung: Einleitung

- **Agil** = wendig, beweglich
- Unbürokratisch, einfach, iterativ
- Kosten- und Zeitersparnis
- 12 Prinzipien
- Agile Methoden
- Agile Prozesse

Agiles Manifest

Beck, Kent, et al.:

"**Individuen und Interaktionen** mehr als Prozesse und Werkzeuge

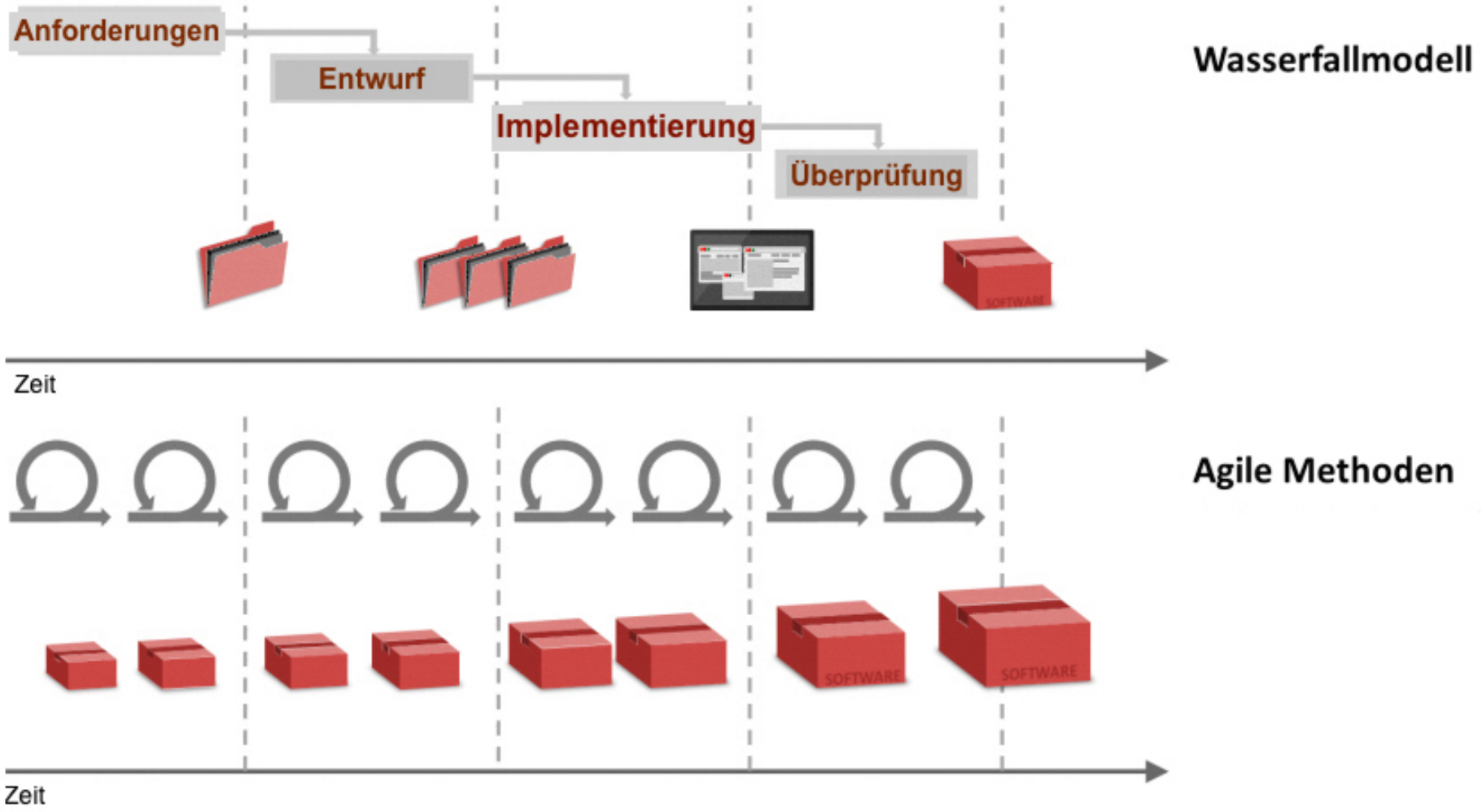
Funktionierende Software mehr als umfassende Dokumentation

Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung

Reagieren auf Veränderung mehr als das Befolgen eines Plans."

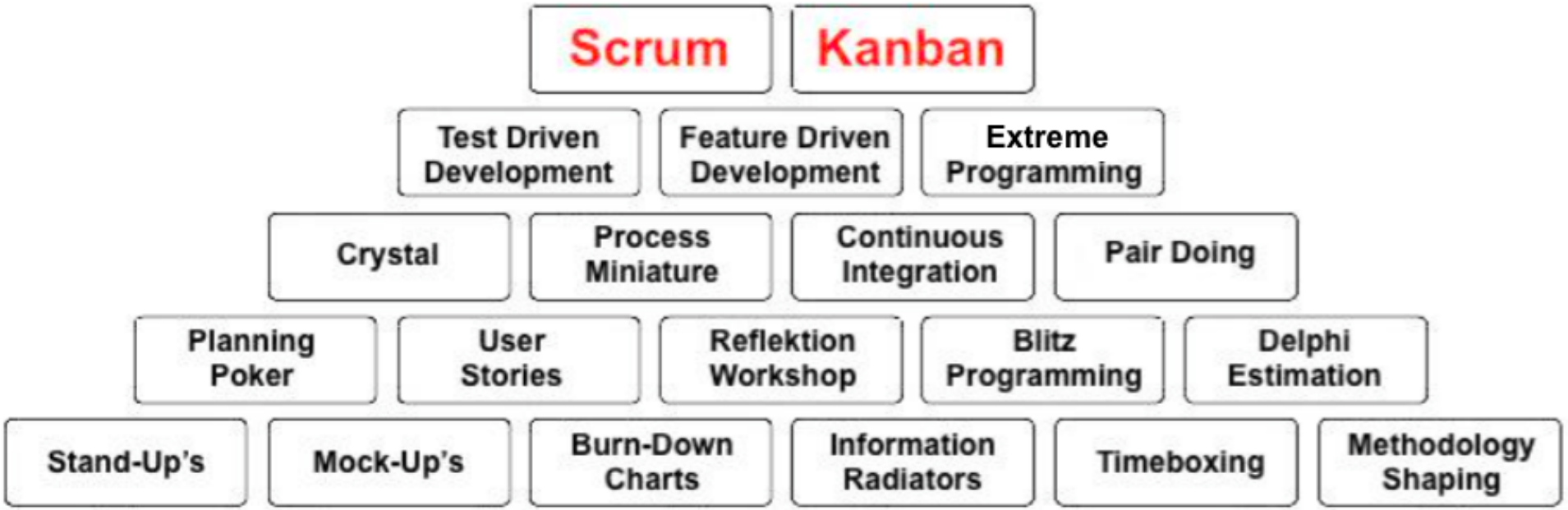
Quelle: Beck, Kent, et al. "Manifesto for agile software development."2001

Wasserfallmodell vs. Agile Methoden



Quelle: http://blog.presentationload.de/effizienter-arbeiten-mit-agilem-projektmanagement/

Agile Methoden

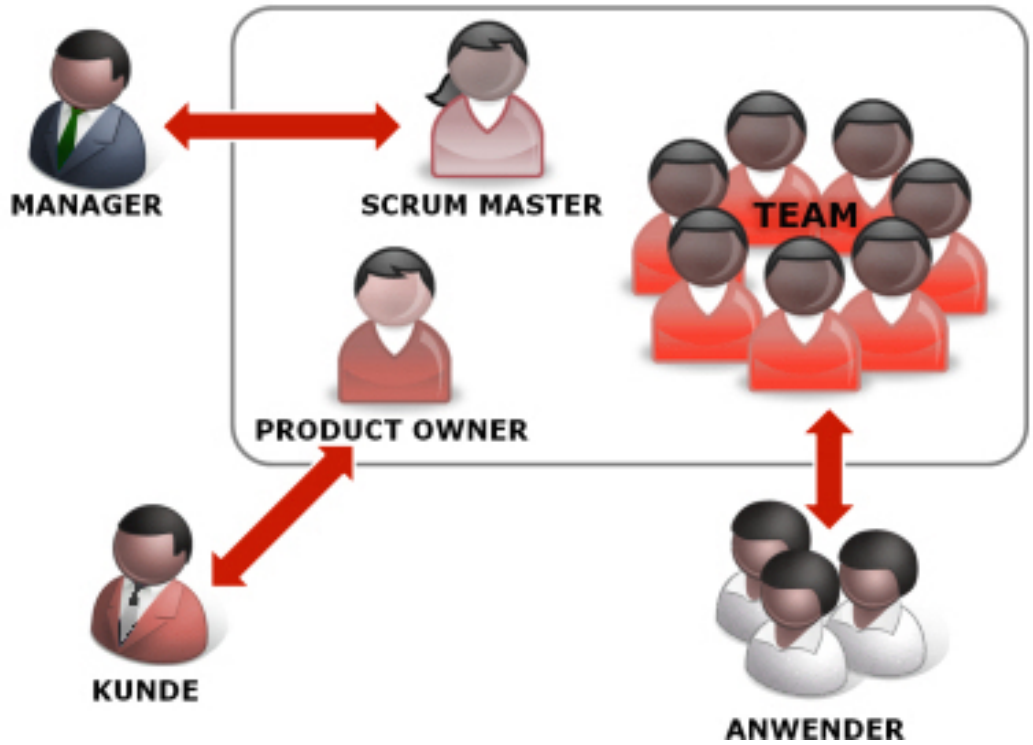


Quelle: http://www.kuglermaag.de/verbesserungskonzepte/agile-in-automotive.html

Sprint
Backlog
Team
ScrumMaster
Standup
Scrum
User-Stories
Sprint-Planning
Anforderungen
Time-to-Market
Software-Entwicklung
Retrospektive
ProductOwner
Review

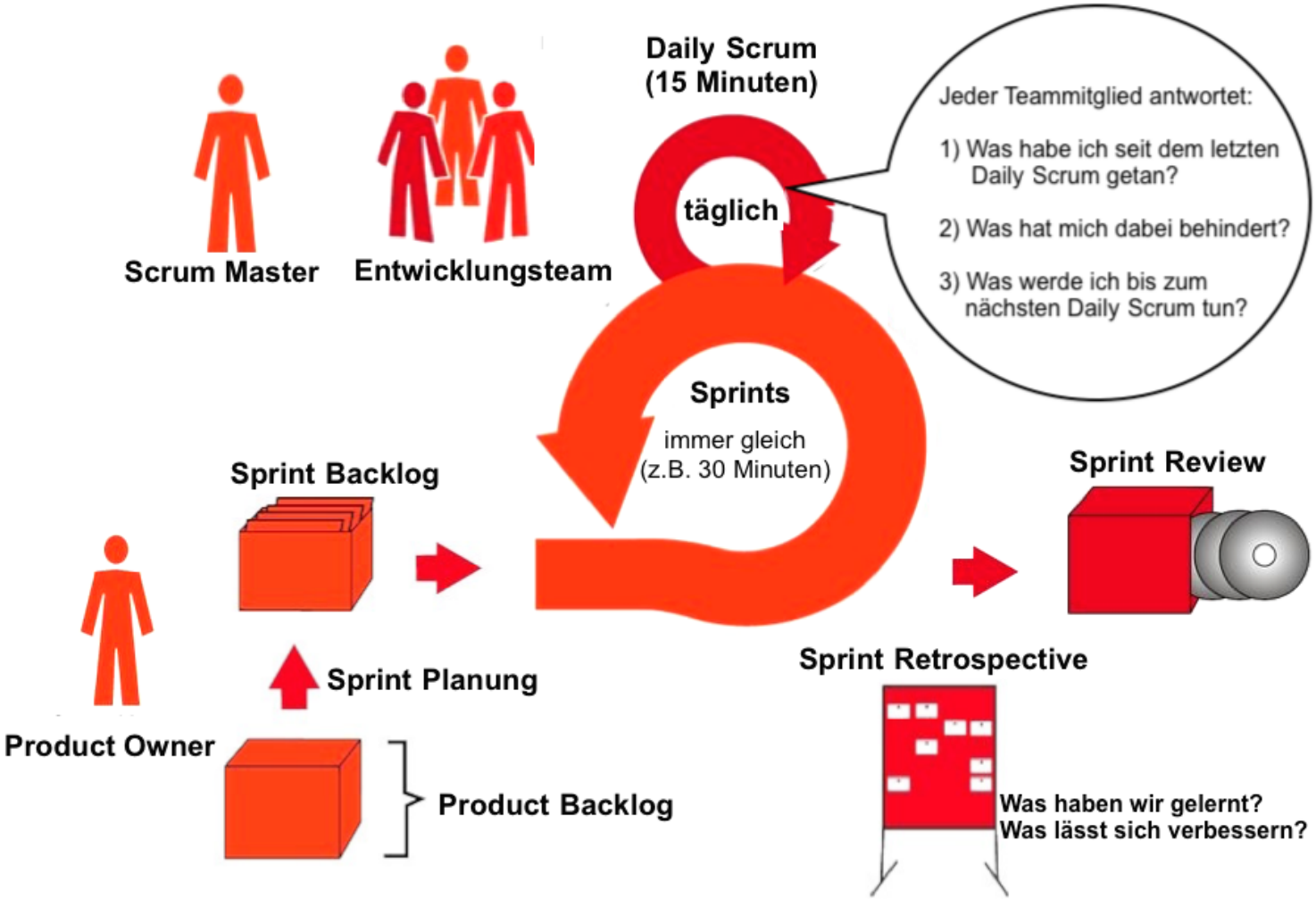
Team (3 Rollen)

- Product Owner
- Entwicklungsteam
- Scrum Master



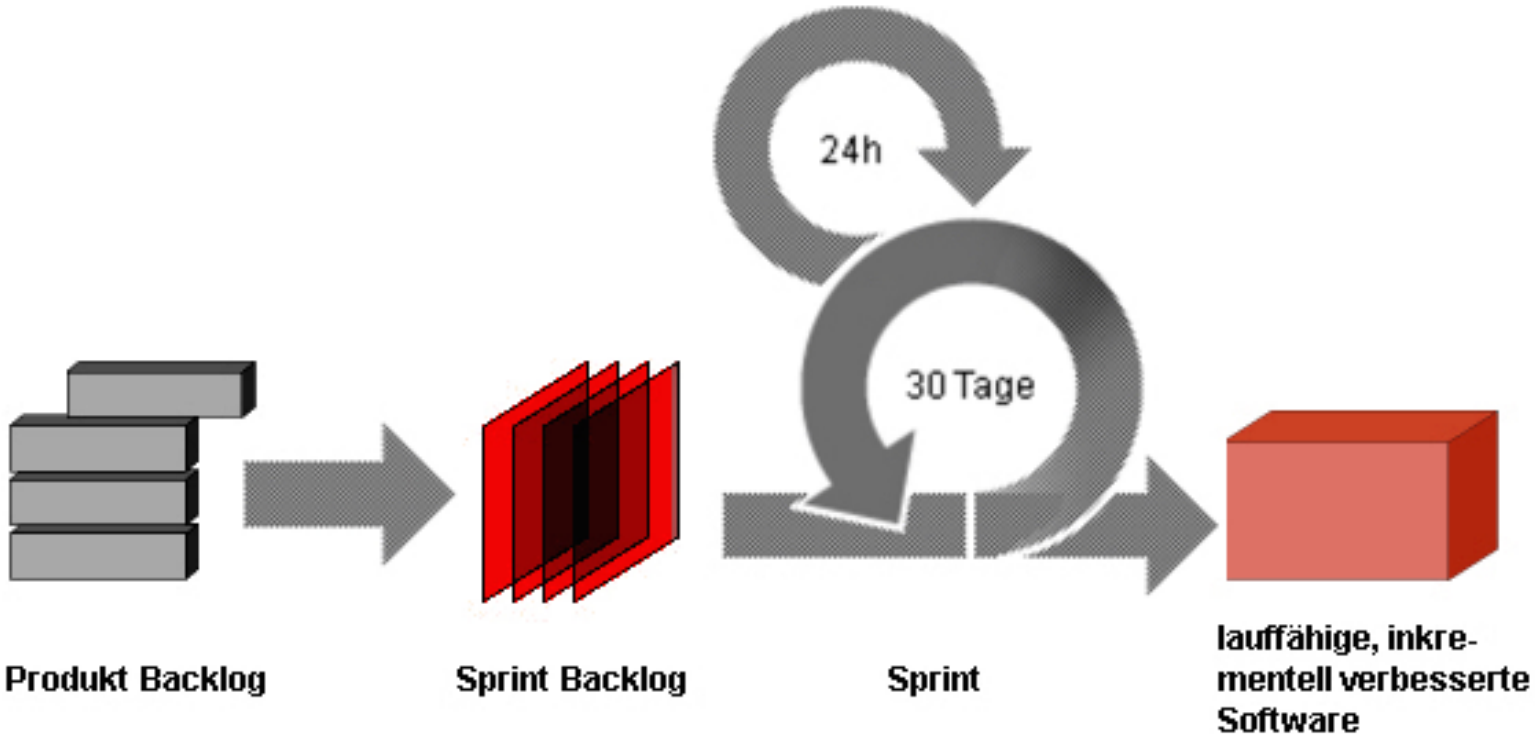
Quelle: http://winf-wiki.fhslabs.ch/images/c/c8/Rollen.png

Scrum im Detail



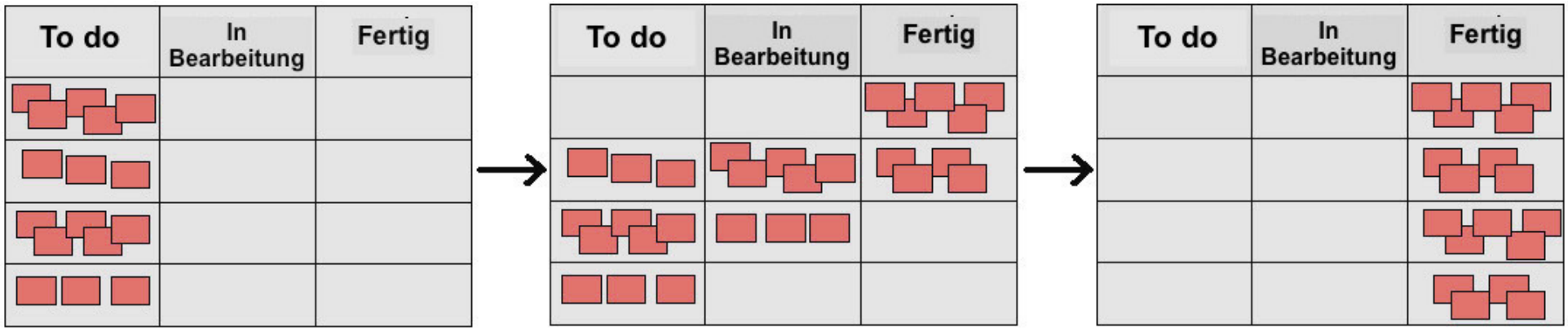
Quelle:http://www.it-agile.de/wissen/methoden/scrum/

Sprint



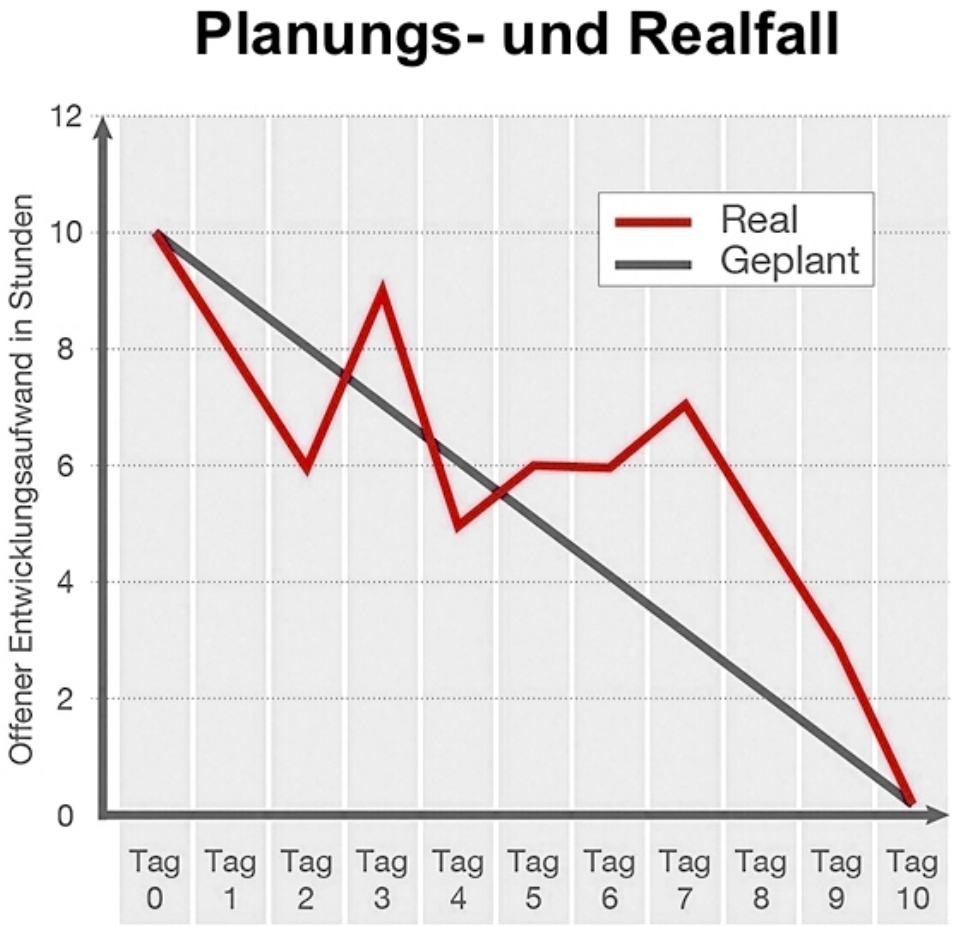
Quelle: https://infos.seibert-media.net/display/Websoftware/Scrum

Taskboard



Quelle: http://if-blog.de/bf/scrum-vs-kanban-1-of-2/

Burndown-Diagramm



Quelle: http://www.scrum-kompakt.de/einfuehrung-in-scrum/fortschrittskontrolle-im-sprint/

Scrum:

Vorteile

- einfach zu lernen
- klare Rollen
- gut strukturierter Entwicklungsprozess

Scrum:

Vorteile

- einfach zu lernen
- klare Rollen
- gut strukturierter Entwicklungsprozess

Nachteile

- hoher Kommunikations- und Abstimmungsaufwand
- Selbstorganisation

Kanban
Agile
Project Management
Kanban
Software
Development
Change
Lifecycle
Complexity
Collaboration
Value
Planning
Team
Kaizen
Manifesto
SDLC
JIT
Principles
Teamwork
Agility
Toyota Way
Deming
Values
Leadership
Engineering
Lean
Systems
TPS
Work
Continuos Improvement
Requirements
Productivity
Code
Zero Defects
Agile 2.0
PMBOK
Processes
Waterfall
Reflection
PDSA
ALM
XP
System Thinking
Working Software
Respect

Kanban: Definition

"Signalkarte"(jap.)

Prinzipien:

- Visualisierung (Tafel)
- Anzahl paralleler Arbeiten (**WiP** = Work in Progress) wird reduziert
- kontinuierliche Verbesserung

Kanban: Definition

"Signalkarte"(jap.)

Prinzipien:

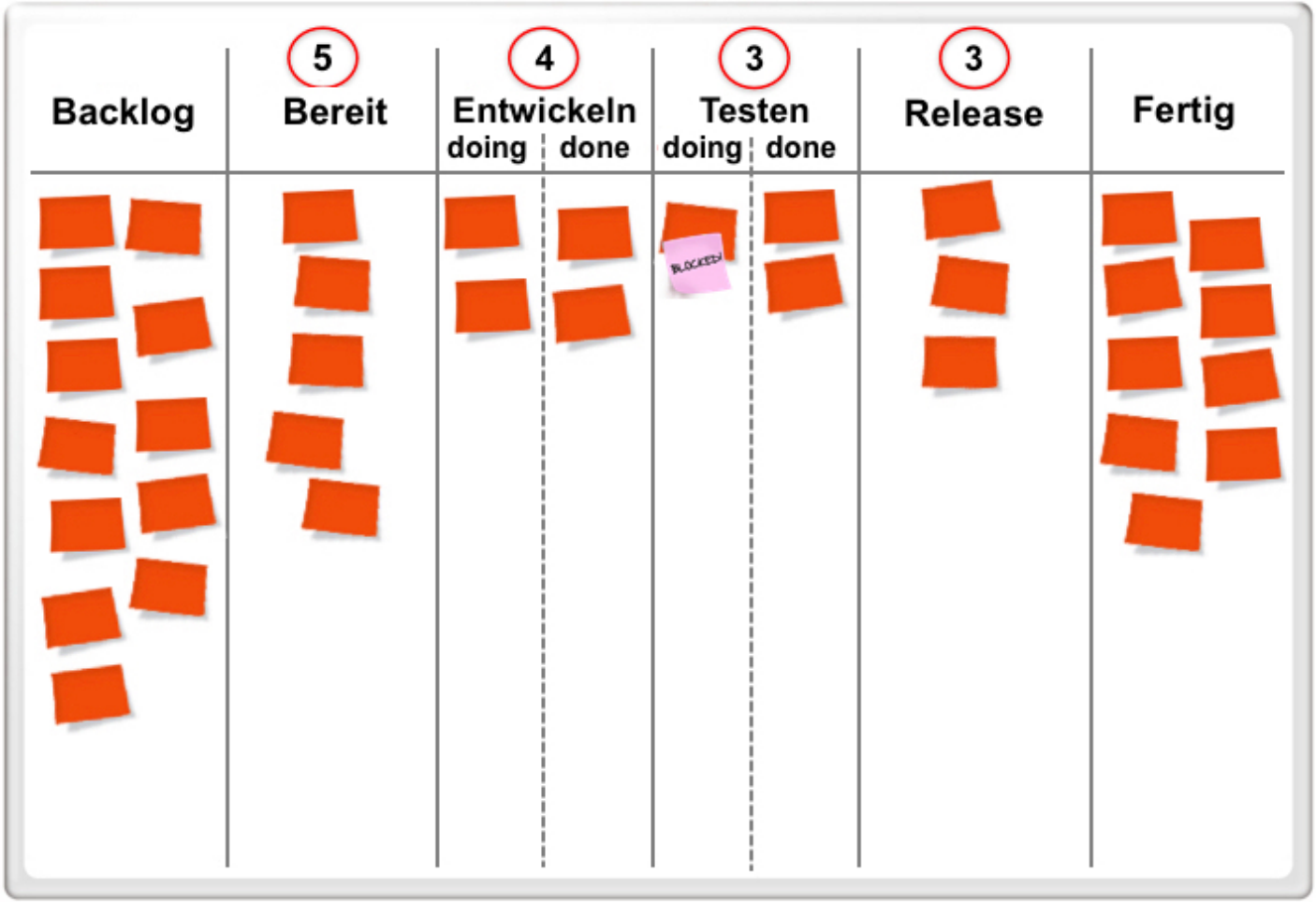
- Visualisierung (Tafel)
- Anzahl paralleler Arbeiten (**WiP** = Work in Progress) wird reduziert
- kontinuierliche Verbesserung

Definition von David Anderson:
Das Kanban-System macht die verschiedenen Prozesse sichtbar und begrenzt die Menge der angefangenen Arbeiten. Der Durchsatz lässt sich mittels Kanban leicht messen und steuern. Prozessregeln werden verdeutlicht. Durch das Verwenden von Modellen werden Verbesserungen direkt sichbar.

Kernpraktiken

- Visualisieren
- Limitierung des WIP
- Den Flow managen
- Genaue Ausführung der Regel
- Einrichten von Feedback-Schleifen
- Im Team verbessern & durch Experimente weiterentwickeln

Kanban: Tafel



Quelle: http://www.it-agile.de/wissen/methoden/kanban/

Kanban:

Vorteile

- Probleme sind **schnell sichtbar**
- auf **jeden** Entwicklungsprozess aufgesetzt werden
- Vermeidung von negatives Multitasking
- Beschreibung der Tasks enthält das Wesentliche

Kanban:

Vorteile

- Probleme sind **schnell sichtbar**
- auf **jeden** Entwicklungsprozess aufgesetzt werden
- Vermeidung von negatives Multitasking
- Beschreibung der Tasks enthält das Wesentliche

Nachteile

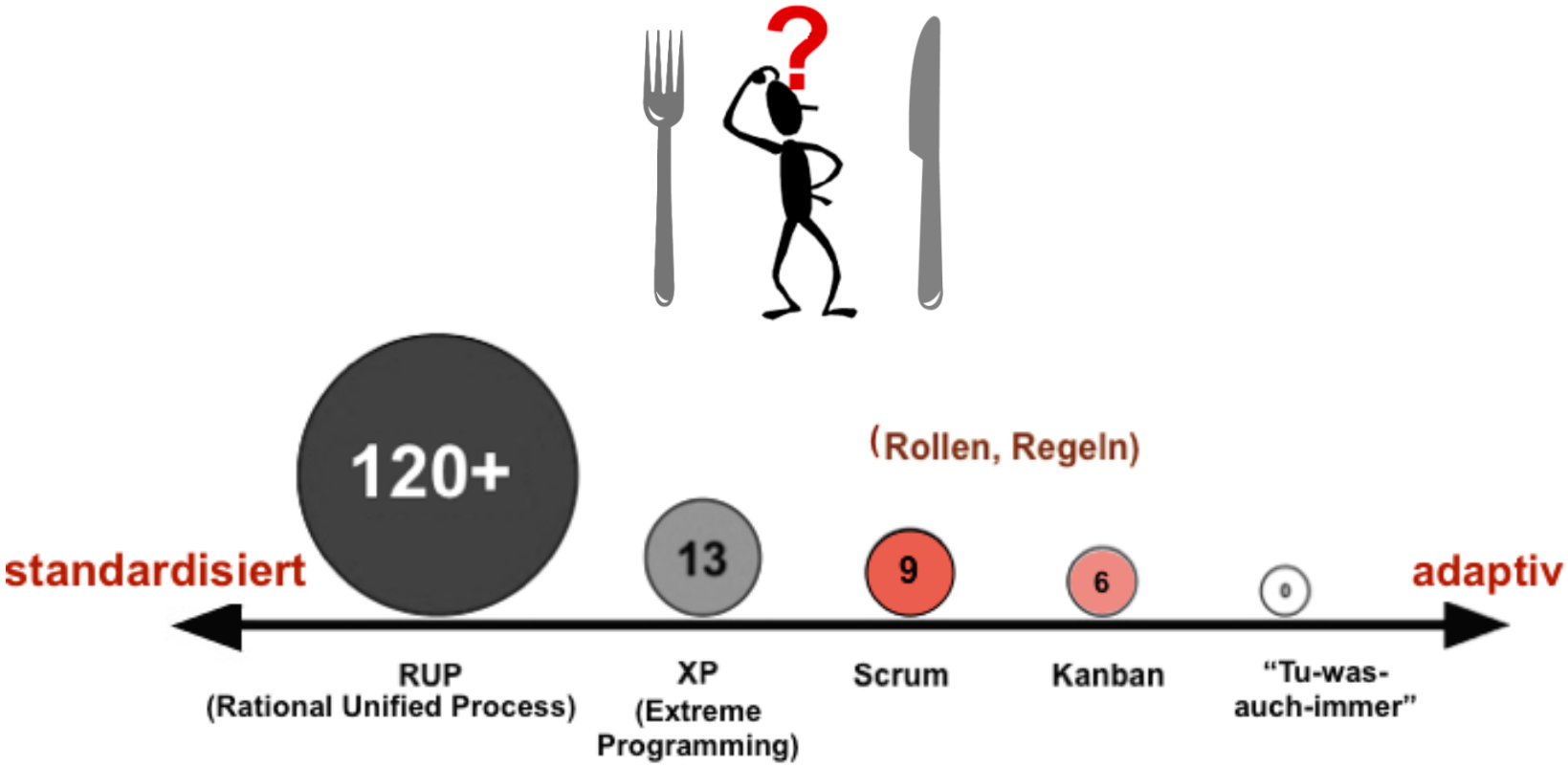
- räumliche Gebundenheit (Teamarbeiten)
- **nicht** beliebig skalierbar (für große Projekte ungeeignet)
- je größer das Team desto unübersichtlicher das Projekt

Allgemeines

- Keine feste Hierarchie innerhalb des Teams
- Eigeninitiative
- keine feste Rollenverteilung
- Eigenverantwortung
- Kompatibilität
- Großen Unternehmen nutzen Kanban als Teil ihres allgemeinen Projektmanagements
- für kleine Teams & kleine Projekte geeignet

Kanban
vs
Scrum
Unterschiede
Ähnlichkeiten
Präskriptiv
Adaptiv
Lean
Agile
Geschwindigkeit
Burndown
optional
WIP-Limit
Durchlaufzeit
Funktionsübergreifende
vorgeschrieben
Standartmetrik
Planung
Prozessverbesserung
Anwendung
Spezialisierte

Scrum oder Kanban?

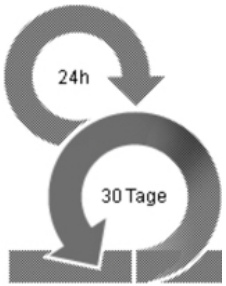


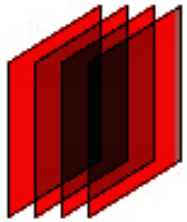
Quelle: Kniberg, Henrik, and Mattias Skarin. Kanban and Scrum-making the most of both. Lulu. com, 2010

Unterschiede

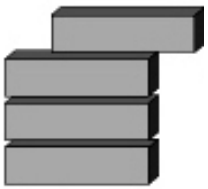
Scrum	Kanban

vorgeschrieben

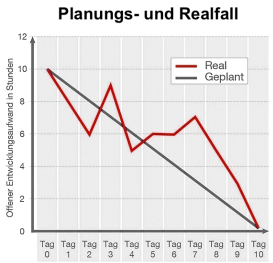


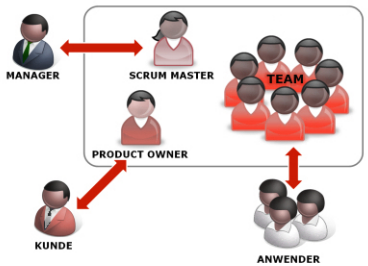


Sprint Backlog



Produkt Backlog



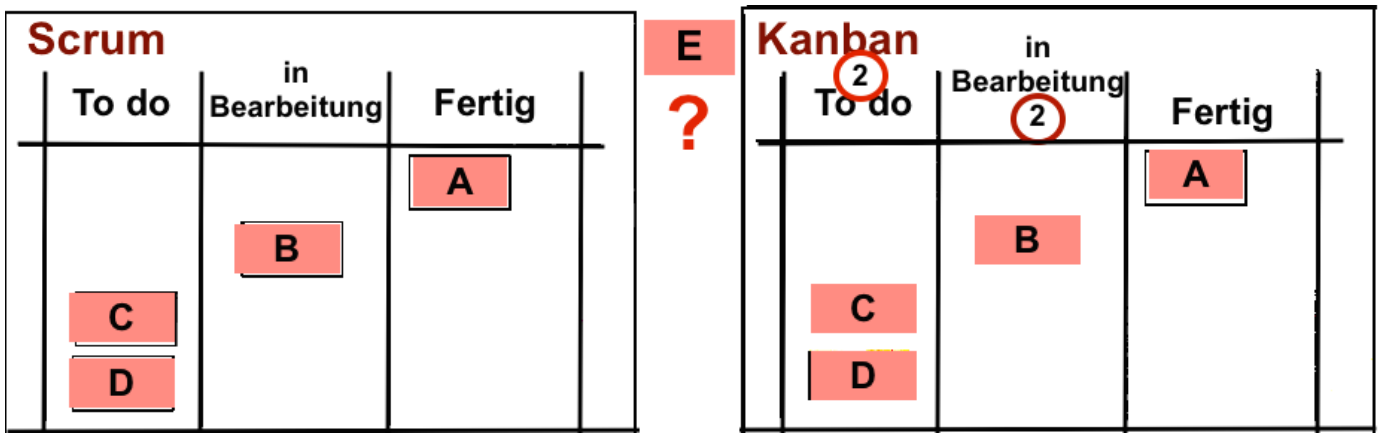


Priorisierung des Product Backlogs
Verpflichtung **Arbeitsaufwand abzuschätzen**
Arbeitspakete müssen **aufgeteilt** werden
Iterationen mit **festem Zeitschema**
Daily Scrum Meetings
Burndown-Diagramm
3 Rollen, **funktionsübergreifende** Teams

optional

Unterschiede

Scrum	Kanban
kann **keine** Arbeitspakete im laufenden Iteration hinzufügen	kann **neue** Arbeitspakete in laufenden Iteration hinzufügen

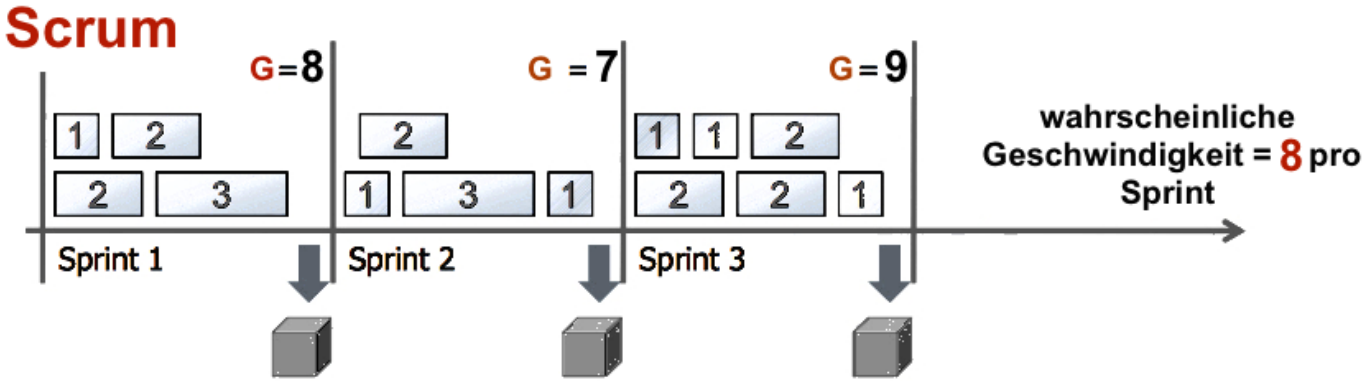


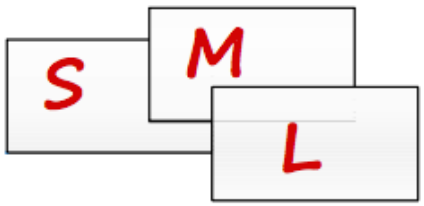
Unterschiede

Scrum	Kanban

Geschwindigkeit
als Standardmetrik
für Planung und
Prozessverbesserung

Durchlaufzeit
als Standardmetrik



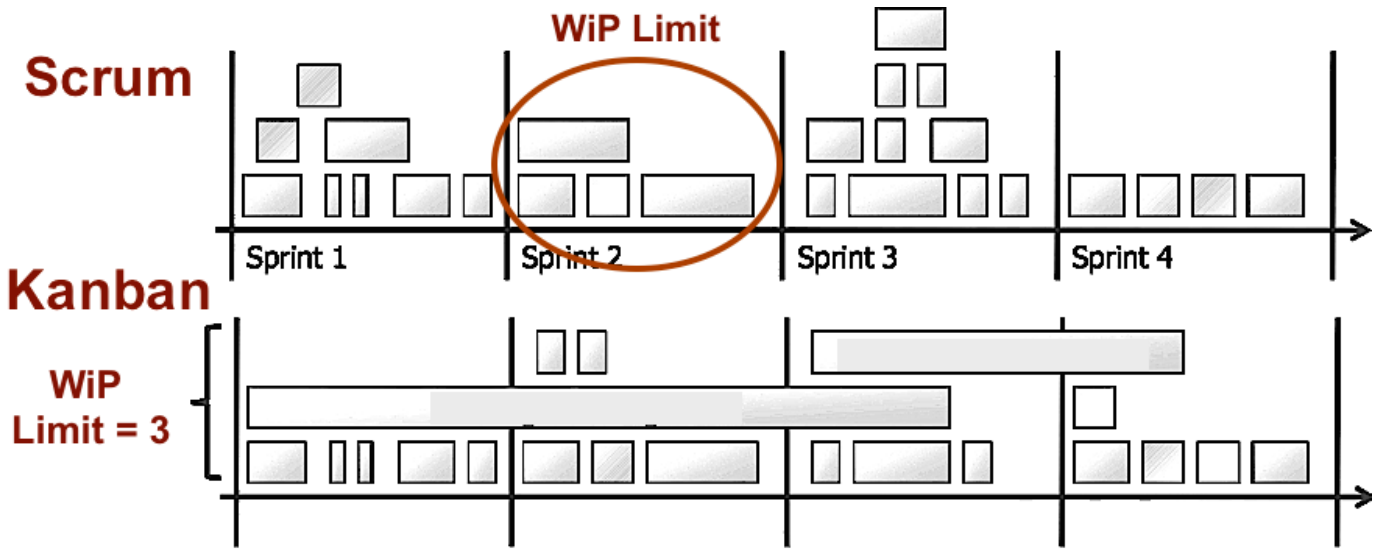


Unterschiede

Scrum | **Kanban**

WiP Limit **indirekt** (durch Sprints)

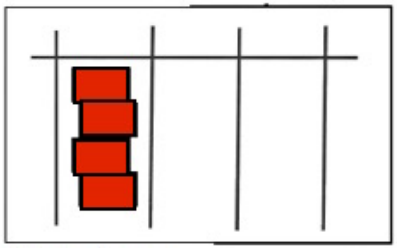
WiP Limit **direkt** (pro Zustand im Arbeitsablauf)



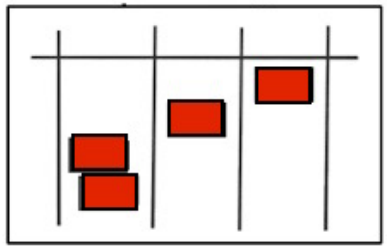
Unterschiede

Scrum	Kanban
Scrum board wird immer **neu eingesetzt**	Kanban board **bleibt** durchgehend **bestehen**

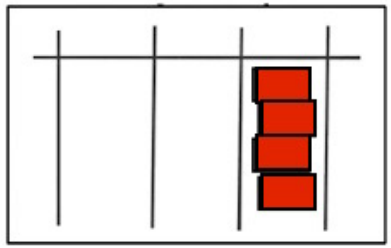
Scrum Anfang



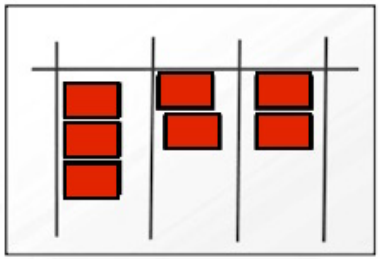
Mitte von Sprint



Ende

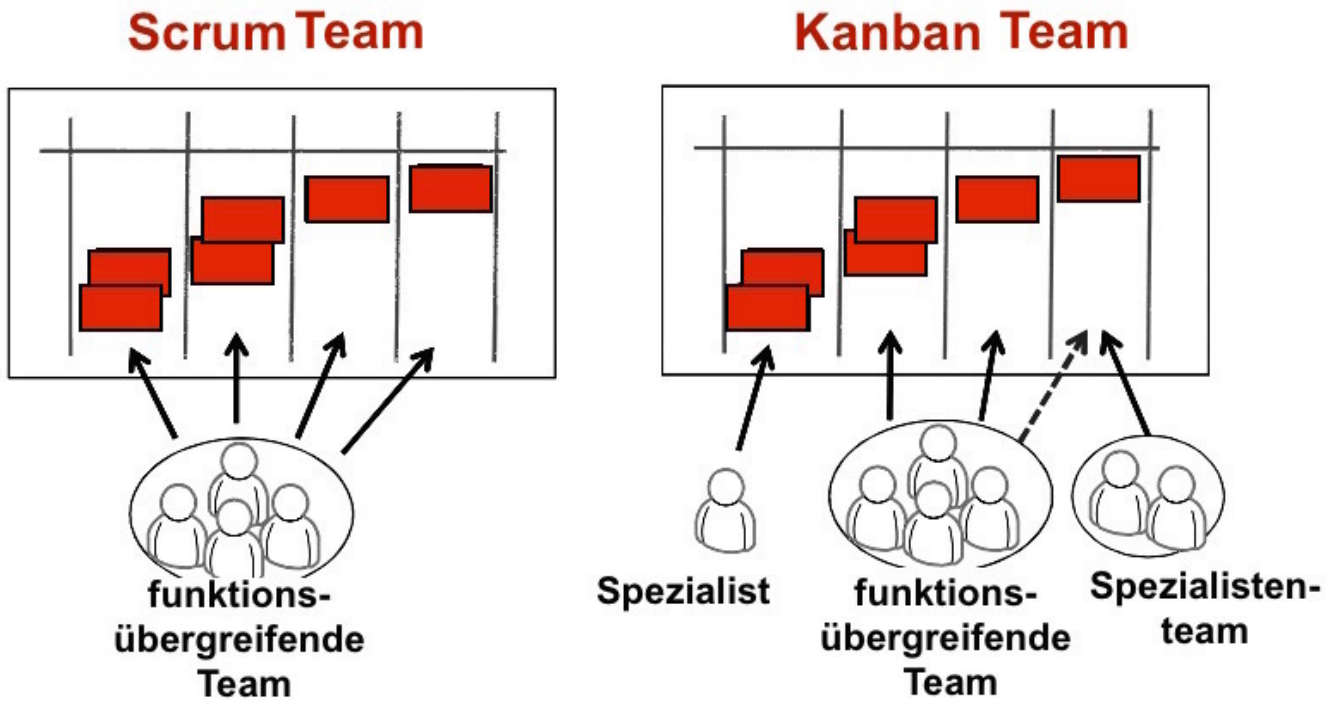


Kanban irgendein Tag



Unterschiede

Scrum	Kanban
Scrum board gehört **einem bestimmten** Team	kann von **mehreren Teams geteilt** werden



Unterschiede

Scrum	Kanban
Anwendung	
In Projekten mit **hoher Komplexität**	in **kleinen** und **nicht komplizierten** Projekten
mit **mehr als 4** Entwicklern	mit **3-4** Entwicklern
Geschwindigkeit spielt eine **große** Rolle	Geschwindigkeit spielt **keine** Rolle
große Projekte, die **strukturiert** sein müssen	wo **Regeln** festzuhalten eine **Zeitverschwendung** ist

Ähnlichkeiten

Scrum und Kanban

- setzen das **Pull-Prinzip** ein
  (Das Team soll erst dann seine Leistung erbringen, wenn diese Leistung von einem Kunden angefordert wird.)

Ähnlichkeiten

Scrum und Kanban

- setzen das **Pull-Prinzip** ein
  (Das Team soll erst dann seine Leistung erbringen, wenn diese Leistung von einem Kunden angefordert wird.)
- begrenzen **WiP**

Ähnlichkeiten

Scrum und Kanban

- setzen das **Pull-Prinzip** ein
  (Das Team soll erst dann seine Leistung erbringen, wenn diese Leistung von einem Kunden angefordert wird.)
- begrenzen **WiP**
- nutzen **Transparenz**

Ähnlichkeiten

Scrum und Kanban

- setzen das **Pull-Prinzip** ein
  (Das Team soll erst dann seine Leistung erbringen, wenn diese Leistung von einem Kunden angefordert wird.)
- begrenzen **WiP**
- nutzen **Transparenz**
- konzentrieren sich darauf, **schnell** und **oft** ausführbare Software **auszuliefern**

Ähnlichkeiten

Scrum und Kanban

- setzen das **Pull-Prinzip** ein
  (Das Team soll erst dann seine Leistung erbringen, wenn diese Leistung von einem Kunden angefordert wird.)
- begrenzen **WiP**
- nutzen **Transparenz**
- konzentrieren sich darauf, **schnell** und **oft** ausführbare Software **auszuliefern**
- basieren auf **selbstorganisierten Teams**

Ähnlichkeiten

Scrum und Kanban

- setzen das **Pull-Prinzip** ein
  (Das Team soll erst dann seine Leistung erbringen, wenn diese Leistung von einem Kunden angefordert wird.)
- begrenzen **WiP**
- nutzen **Transparenz**
- konzentrieren sich darauf, **schnell** und **oft** ausführbare Software **auszuliefern**
- basieren auf **selbstorganisierten Teams**
- der Releaseplan wird auf Basis **empirischer Daten** kontinuierlich verbessert

Danke für Ihre Aufmerksamkeit!
Fragen?

Quellen:

- http://www.it-agile.de
- http://www.scrum-kompakt.de
- http://agilemanifesto.org/iso/de/
- Beck, Kent, et al. Manifesto for agile software development, 2001.
- http://www.scrumguides.org
- https://www.scrumalliance.org
- http://kanbanblog.com/explained/
- Anderson, David J. Kanban. Blue Hole Press, 2010.
- Kniberg, Henrik, and Mattias Skarin. Kanban and Scrum-making the most of both. Lulu. com, 2010.