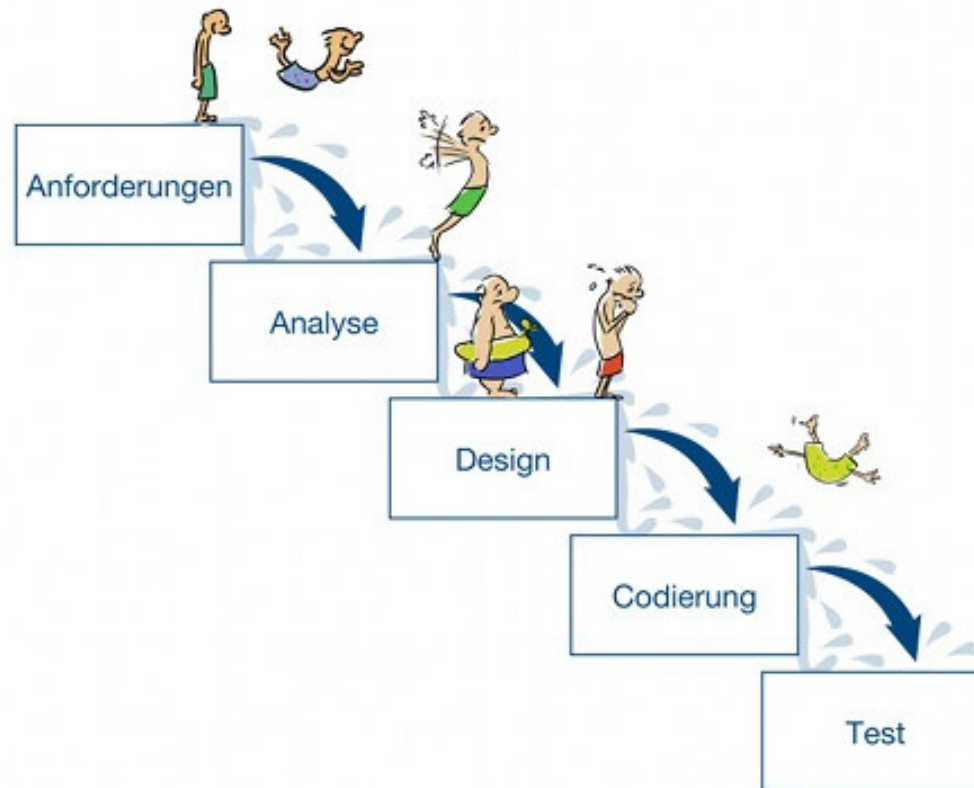


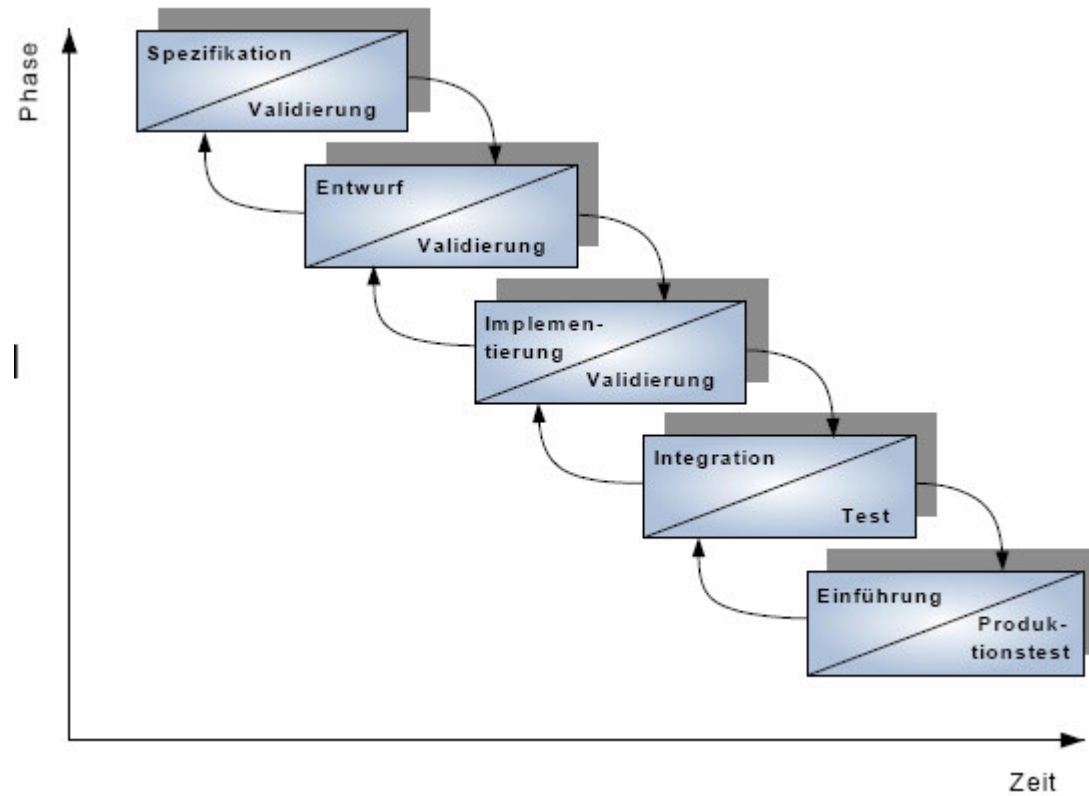
## Agiles Vorgehen und SCRUM



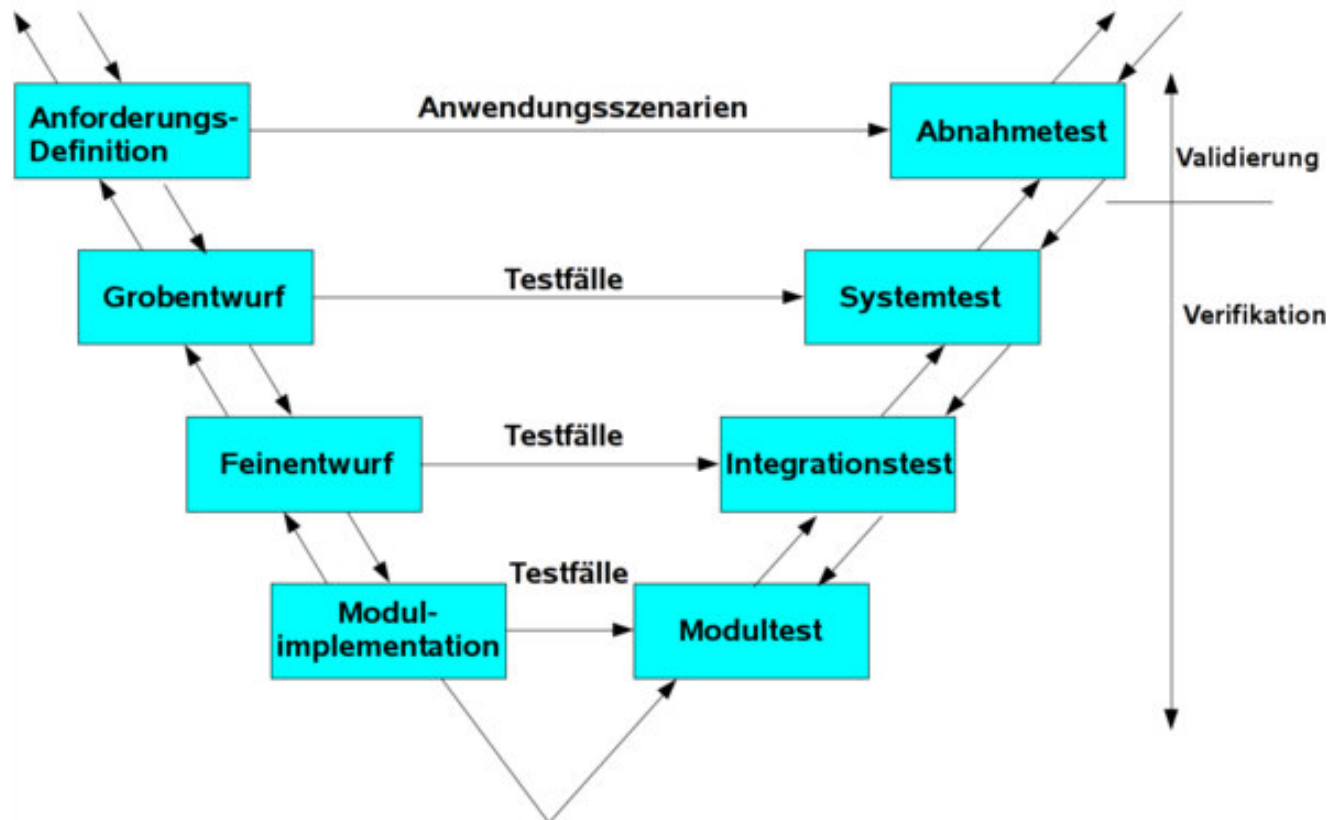
### Klassisches Modell



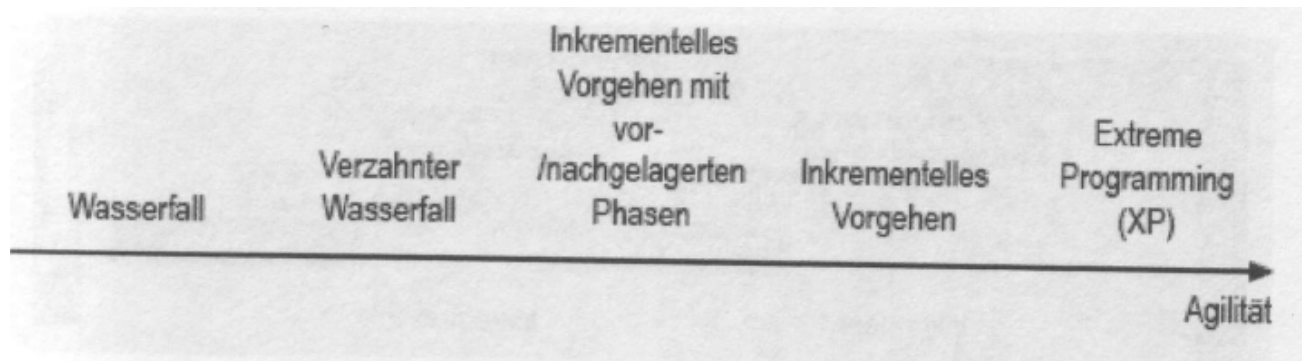
### Wasserfallmodell



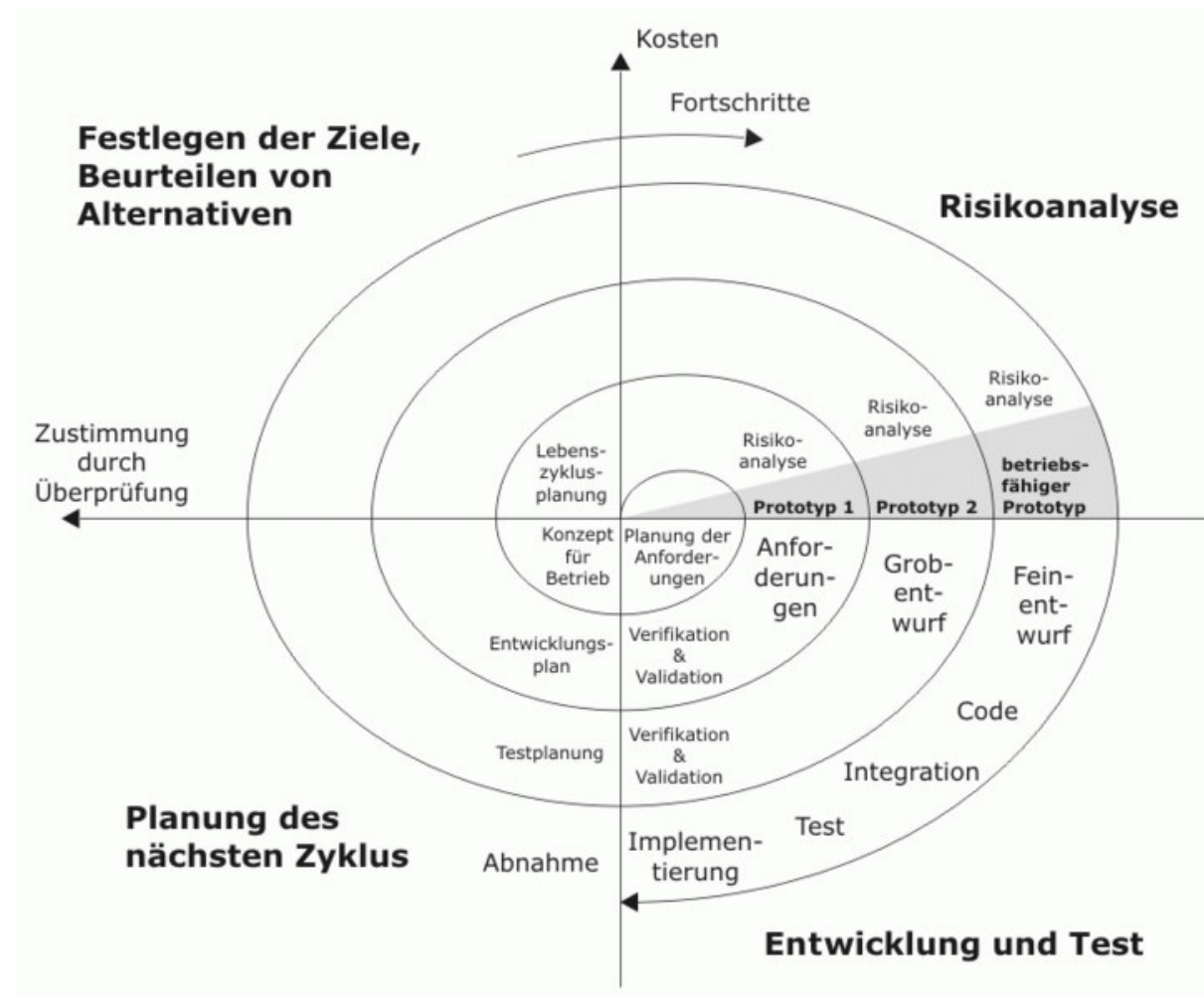
### V-Modell



### Agilität



### Spiral Modell



# Warum aber nun „agil“?

### Was war bisher in der Projekten?

- Wiederkehrende Arbeiten
- Vorhersagbare Lösungen
- Führungsstil in der Regel autoritär/überwachend

### Die Arbeit im 21. Jahrhundert ist aber anders

- Wissensbasiert und komplex
- Nicht von Anfang an vollständig erfassbar
- Oft nur durch die Zusammenarbeit in einem Team lösbar

⇒ Bisherige Ansätze sind nicht mehr .....

⇒ Wir brauchen eine neue Art der ..... &  
.....

# Geschichte der Agilen Methoden

- Erste Ansätze gab es bereits Anfang der 90-iger Jahr
- Auslöser war der Standish Group Chaos Report 1994
- Die Software-Entwickler Jeff Sutherland und Ken Schwaber überlegten wie man Software besser entwickeln könnte
- Die Lösung sollte eine höhere Flexibilität bei der Entwicklung sein
- 1999 veröffentlichten Kent Beck und Andere das Buch „Extreme Programming“
- Im Februar 2001 trafen sich 17 erfahrene Software-Entwickler und erstellten und unterzeichneten das „Manifesto for Agile Software Development“



# Das Agile Manifest

Wir entdecken Wege, Software besser zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Arbeit haben wir Folgendes zu schätzen gelernt:



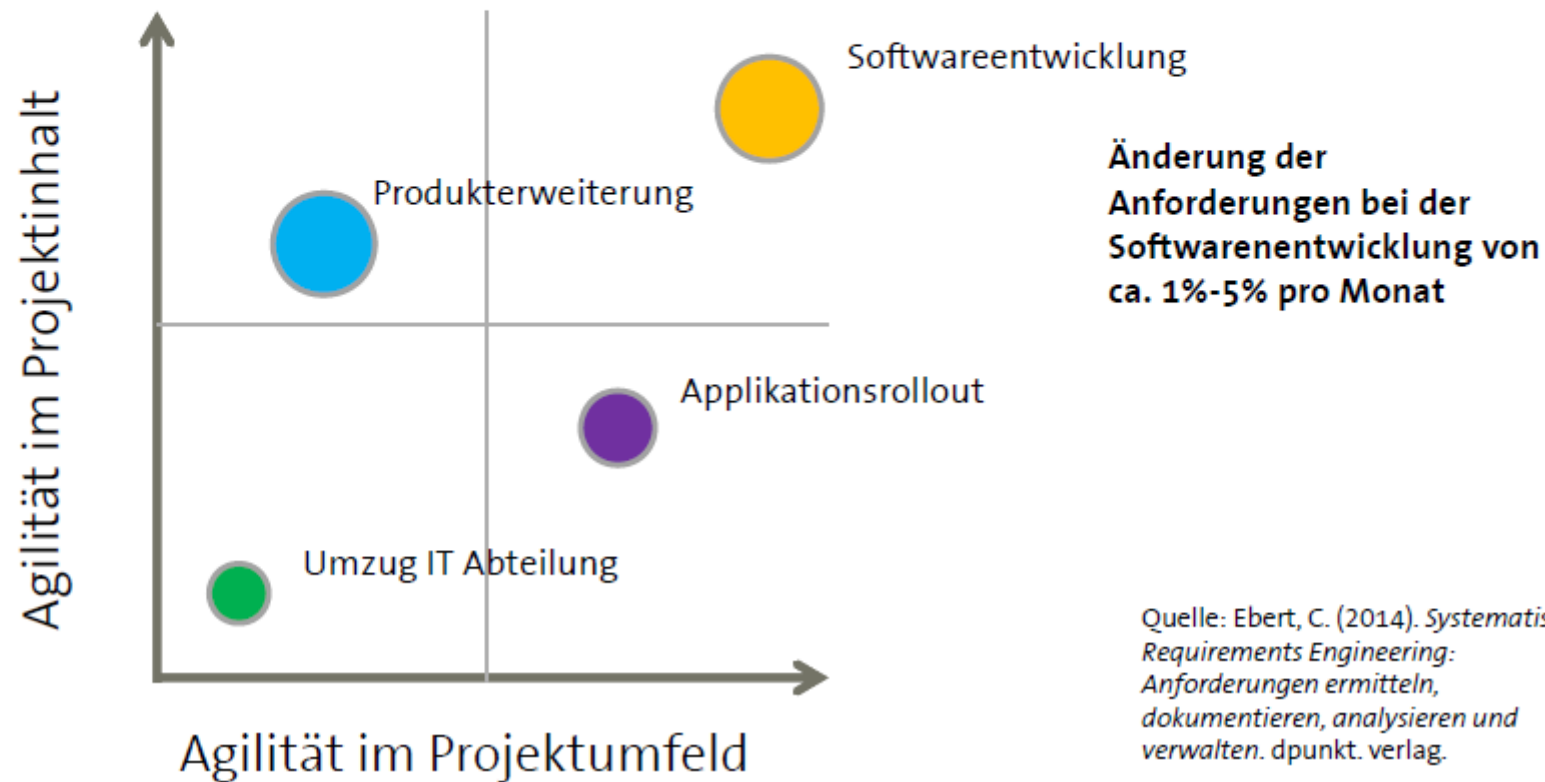
- **Individuen und Interaktionen** sind wichtiger als Prozesse und Werkzeuge.
- **Funktionierende Software** ist wichtiger als umfassende Dokumentation.
- Die **Zusammenarbeit mit dem Kunden** ist wichtiger als Vertragsverhandlungen.
- **Sich auf unbekannte Änderungen einzustellen**, ist wichtiger als einem Plan zu folgen.

Wir schätzen aufgrund unserer Erfahrung die Punkte auf der rechten Seite, aber wir bewerten die Punkte auf der linken Seite höher.

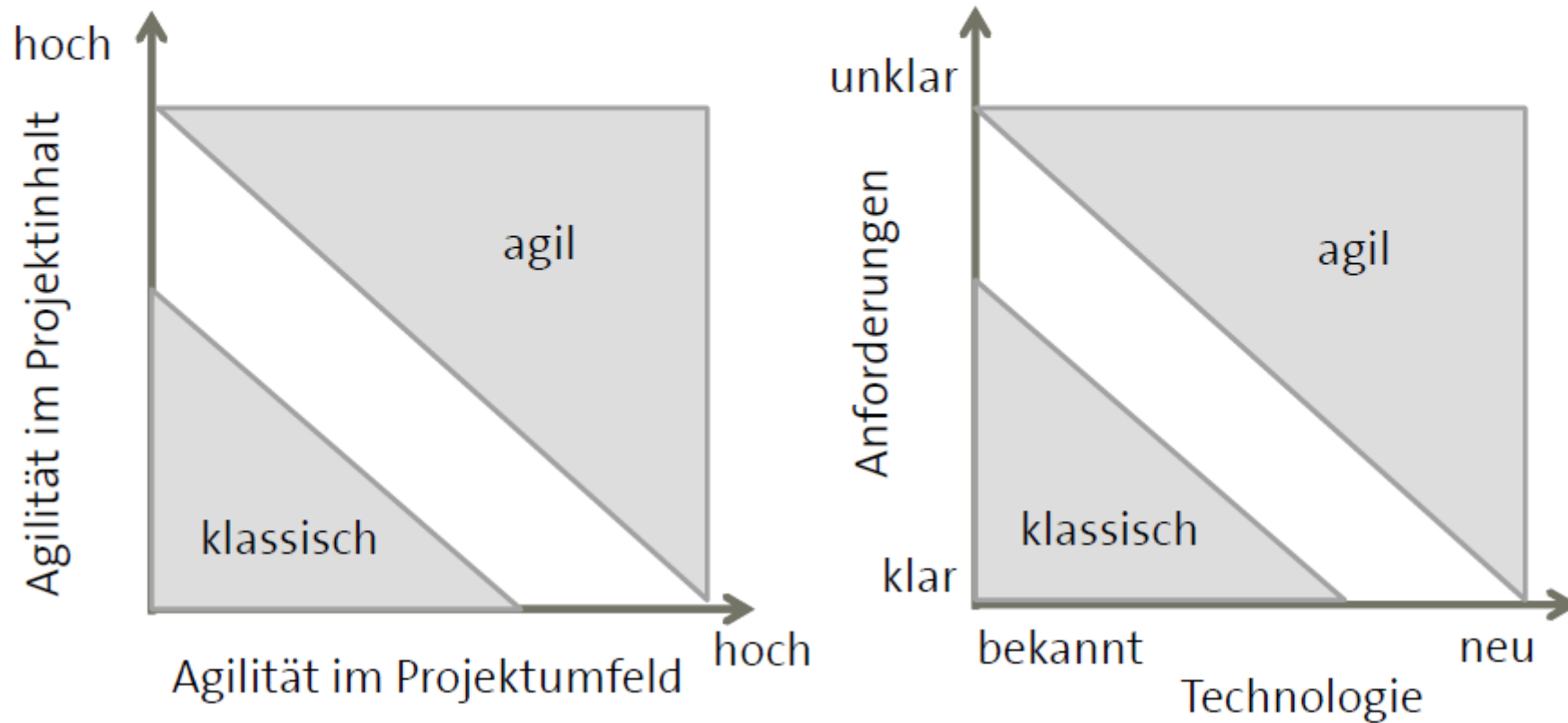
[www.agilemanifesto.org](http://www.agilemanifesto.org)

# Vorgehensmodelle

## Agilität?... Warum ?



# Agiles Projektmanagement Für welche Projekte?



### Agile Prinzipien

- Durch kontinuierliche Auslieferung von für den Kunden „wertvolle“ Software wird die Zufriedenheit des Kunden gesteigert
- Lieferung von „funktionierender“ Software in regelmäßigen und kurzen Zeitspannen
- Enge, tägliche Zusammenarbeit von Fachexperten und Entwicklern während der Entwicklungsphase (Sprint)
- Motivation der Mitarbeiter durch ein optimales Umfeld und Unterstützung
- Informationsaustausch möglichst von Angesicht zu Angesicht
- Der wichtigste Projektfortschritt ist eine funktionsfähige Software
- Selbstorganisation des Teams bei der Planung und Umsetzung innerhalb des Sprints
- Selbstreflektion des Teams

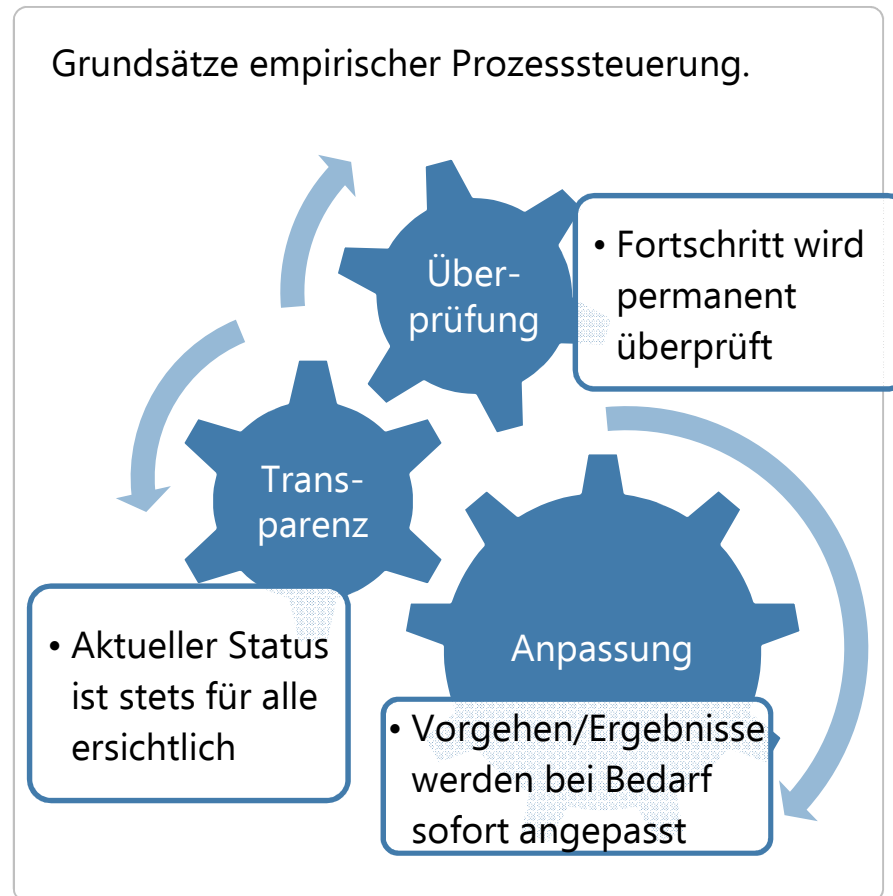
### Agilität

- Nach 40 Jahren Softwaretechnik hat sich der relative Anteil der erfolgreichen, problematischen und total abgestürzten Projekte nicht signifikant verschoben
- Andererseits: Softwareprojekte sind heute um Größenordnungen komplexer als vor 40 Jahren
- Methodischen Fortschritte durch Komplexitätszuwachs „verbraucht“
- Andere Sichtweise: Die eingesetzten Prozesse fokussieren nicht auf das, was eigentlich wichtig ist

# Was ist Nun Scrum?

### Scrum.

- Framework .....
- Empirisch, .....
- Kein vollständiges, .....
- Aber: Mindeststandards  
➔ "Scrum Guide", "Scrum Primer"



# Nutzen von Scrum

### Projekterfolg: Wasserfall vs. agile Entwicklung

Wasserfall:



Agil:



Quelle: The CHAOS Manifest, The Standish Group, 2012

### Gründe für Scrum

- Etabliertes Vorgehen
- Erhöhte Flexibilität in der Projektdurchführung
- Verbesselter Kontakt zum Kunden
- Erfolgreichere Projekte
- Wird von Kunden explizit gefordert
- Erhöhte Mitarbeitermotivation durch eigenverantwortliches Arbeiten im Team



### Scrum Facts

- Scrum ist ein agiles Managementframework
- Scrum ist nicht auf Softwareprojekte limitiert...
- Es enthält keine Praktiken, die vorschreiben, wie die Software entwickelt werden soll
- Scrum ist von den Ideen des leanmanagement/leanproductioninspiriert
- Erstes Scrum-Projekt war 1993
- Einer der Väter ist Jeff Sutherland

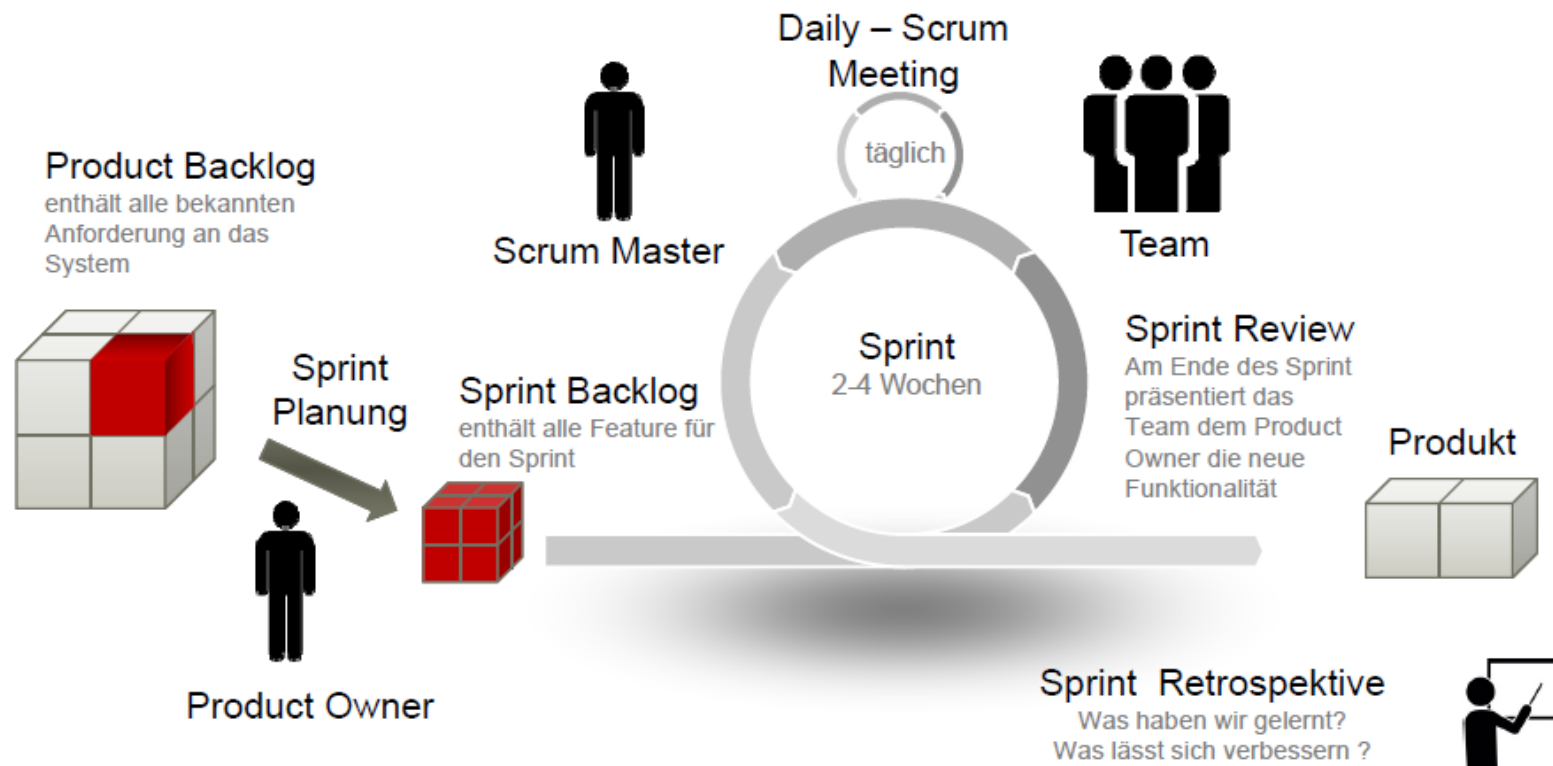




### Was will man mit Scrum erreichen?

- Mitarbeiterzufriedenheit
- .....
- Time to market
- .....
- Produktivität

### Scrum Überblick



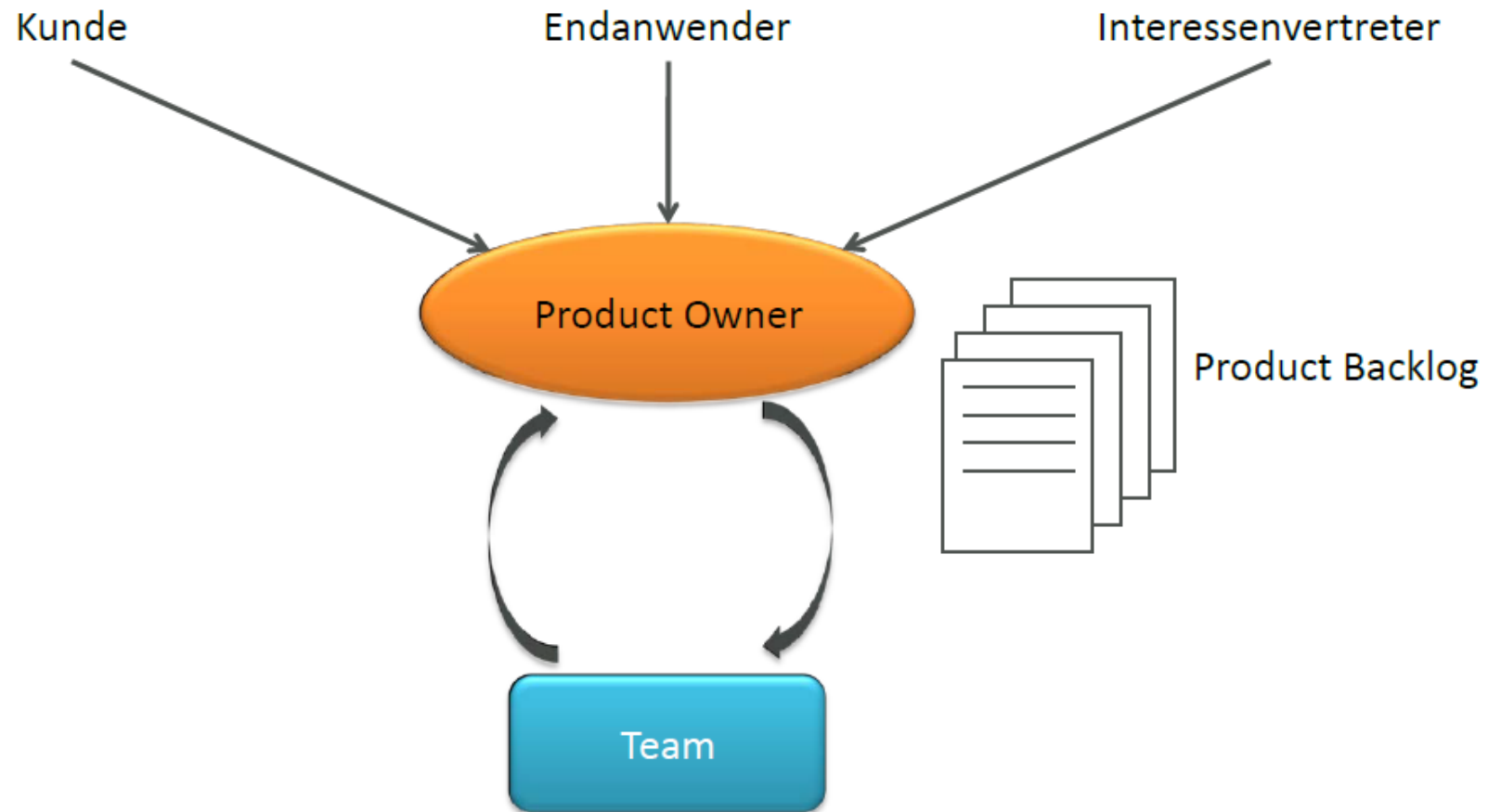
# Scrum Framework

Rollen	Aktivitäten	Artefakte
<ul style="list-style-type: none"><li>• Product Owner</li><li>• Scrum Master</li><li>• Team</li></ul>	<ul style="list-style-type: none"><li>• Sprint-Planung</li><li>• Sprint-Review</li><li>• Sprint-Retrospektive</li><li>• Daily Scrum-Meeting</li><li>• Product Backlog Refinement</li></ul>	<ul style="list-style-type: none"><li>• Product Backlog</li><li>• Sprint Backlog</li><li>• Product Increment</li></ul>

### Artefakt: Product Backlog

- Enthält die Liste aller Anforderungen an das Produkt
- Nicht fix, sondern ein ..... Dokument
  - Entwickelt sich iterativ und inkrementell
  - Scrum kennt keine ..... dennoch ist es oft nützlich Änderungen am Product Backlog zu dokumentieren
  - Alle Einträge werden priorisiert und bzgl. ....
- Das Product Backlog ist möglichst ....., dennoch werden alle wichtigen Anforderungen von Beginn an festhalten.
  - Initial mindestens Anforderungen für die ersten .....
  - Zentrale Funktionalität ins Product Backlog, aber auch ..... Anforderungen

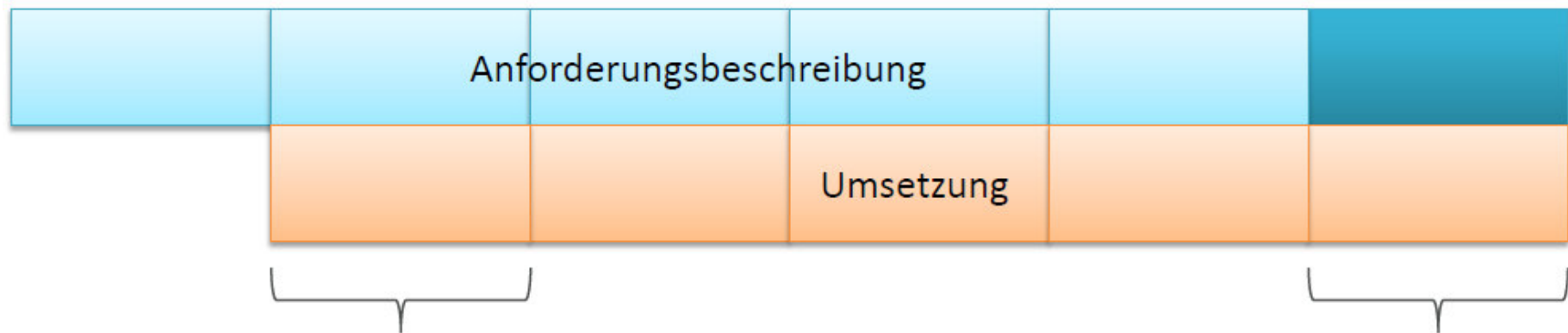
# Erstellen von Anforderungen



### Erstellen von Anforderungen



Traditionelles Vorgehen



Sprint 1

Vorgehen in Scrum

Sprint n

# Vom Epic zur Task

### **Epic:**

Epics sind große User Stories, die weder vernünftig geschätzt noch innerhalb eines Sprints entwickelt werden können (auch Gruppierung von Themen)  
z.B. Benutzerverwaltung, Rechnungserstellung...

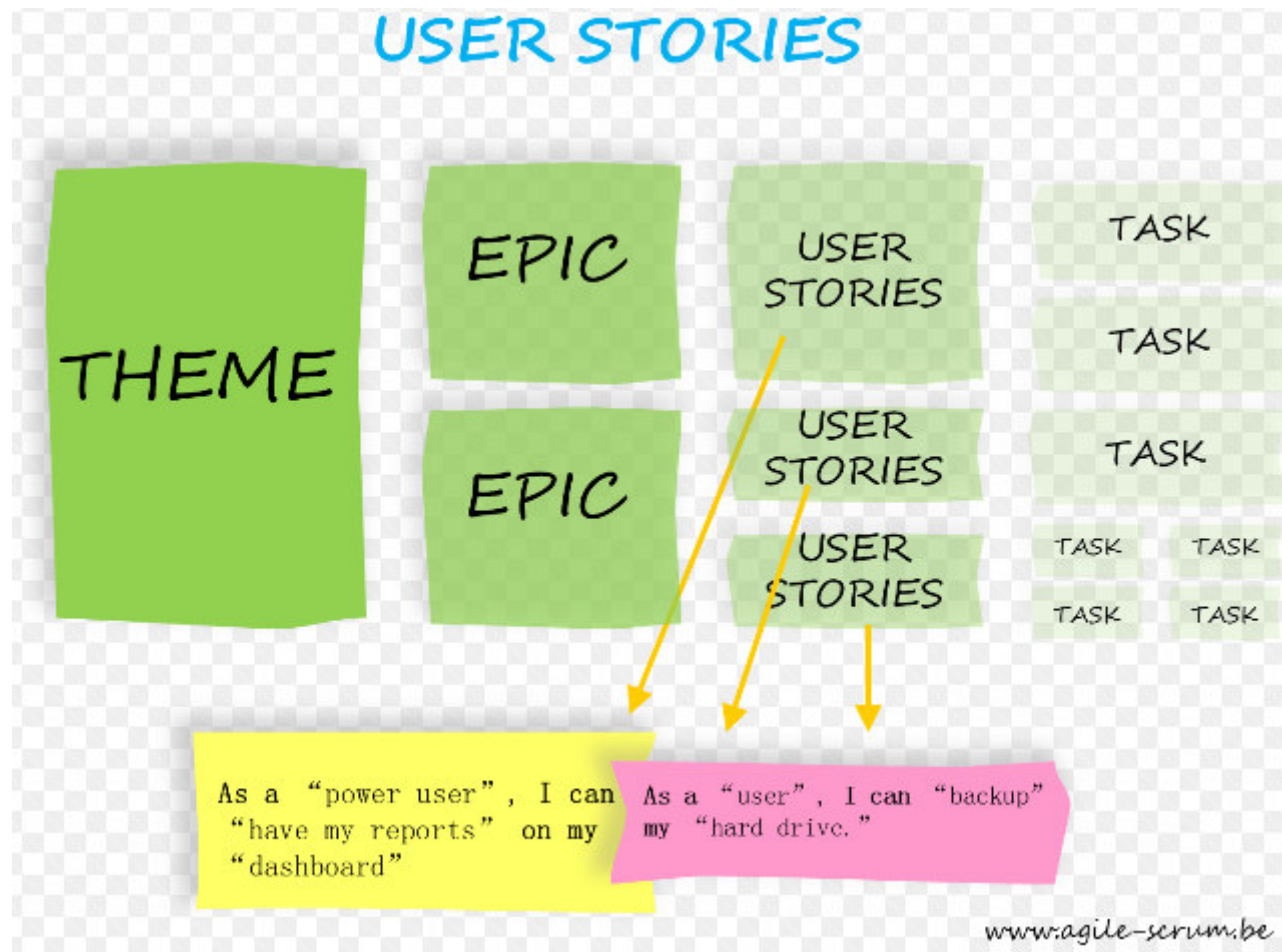
### **User Story:**

Epics werden vom Product Owner in einzelne User Stories herunter gebrochen. Diese sind dann bereits konkret  
z.B. Benutzer erfassen, Benutzer löschen, ...

### **Task:**

Die unterste Ebene bilden die Tasks. Eine User Story wird in verschiedene Tasks aufgeteilt. Diese sind kleine überschaubare Aufgaben, welche nicht länger als einen Tag dauern dürfen und sehr konkret sind  
z.B. Datenbank anlegen, Eingabemaske erstellen, ....

# Anforderungen herunterbrechen





# User Stories sind die Anforderungen

**Sie beschreiben Anforderungen aus Sicht des Endanwenders**

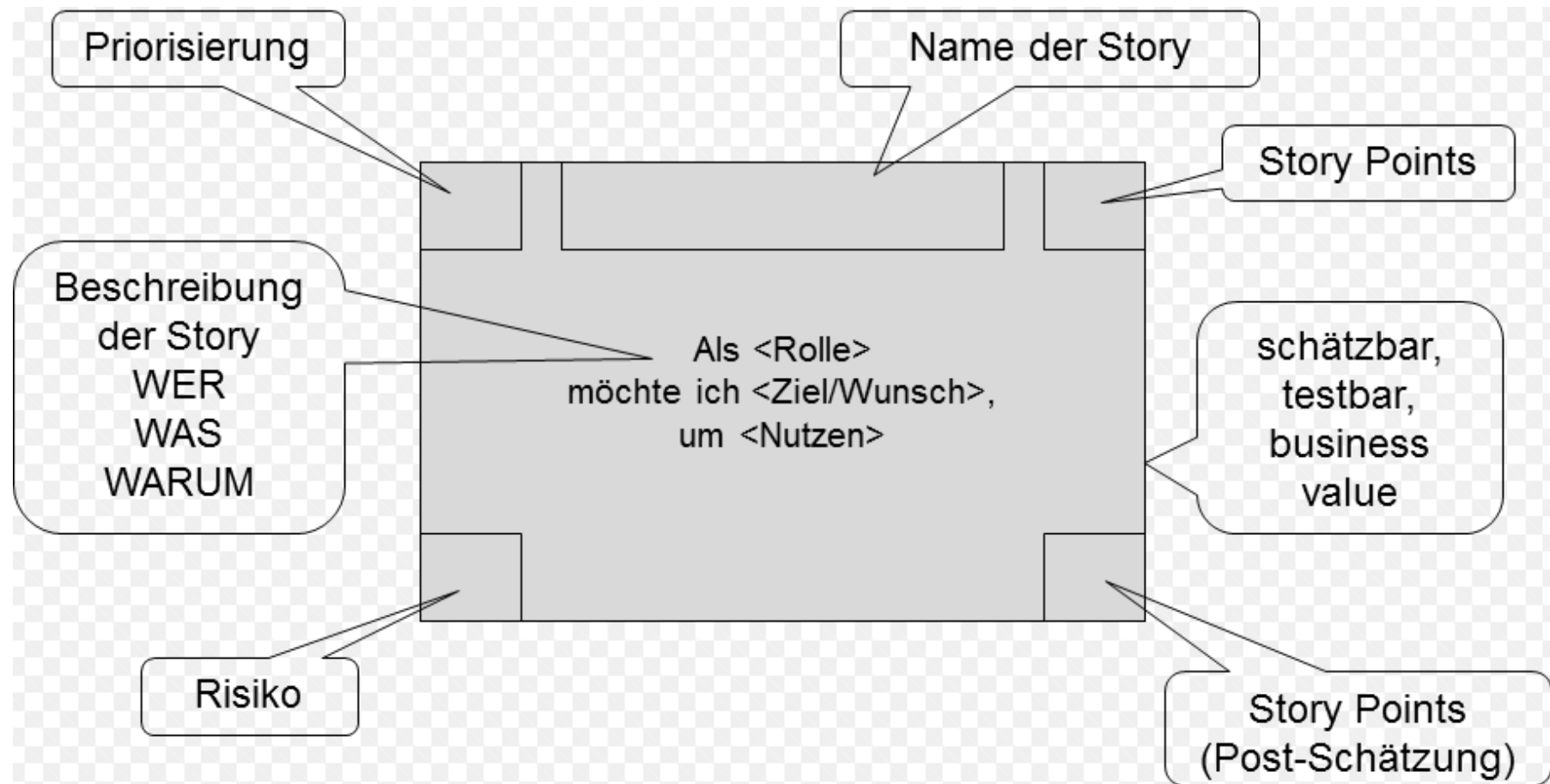
➤ Bestandteile

- .....
- Kurze Beschreibung .....
- .....

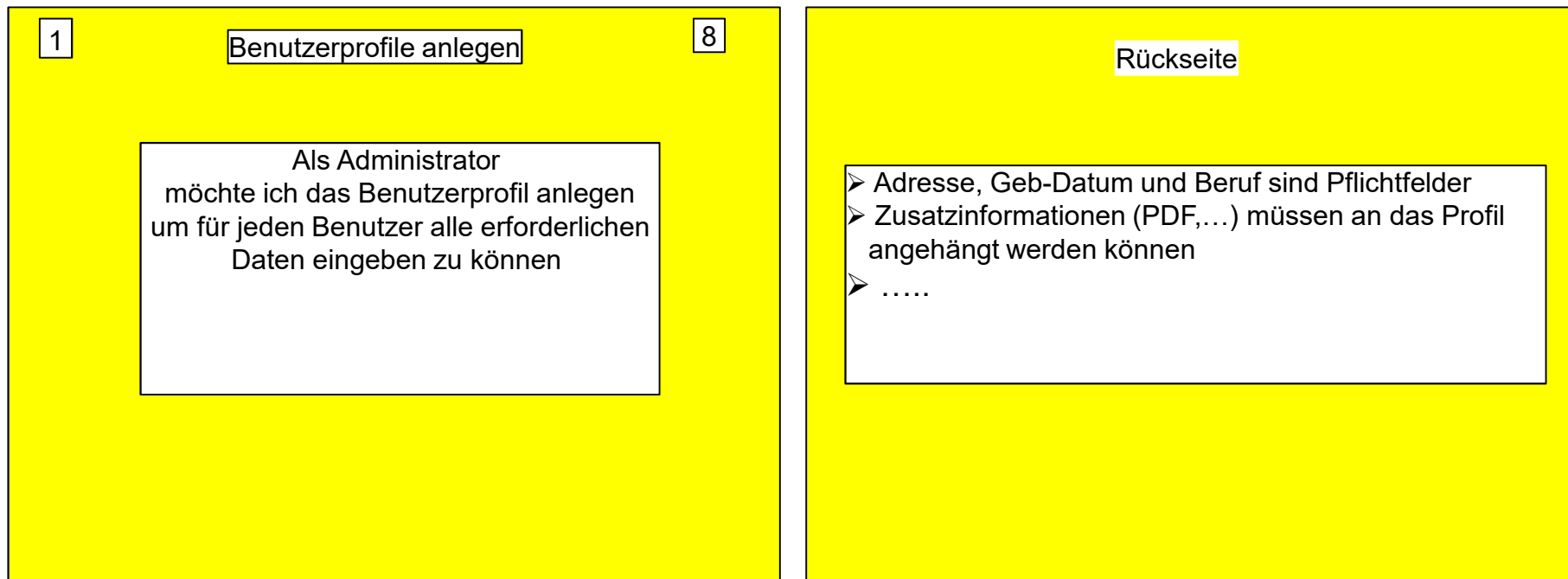
➤ Text beinhaltet Benutzerrolle, ..... und optional den ..... es Merkmals

➤ 3 C-Bedingung einhalten: Card, Conversation, Confirmation

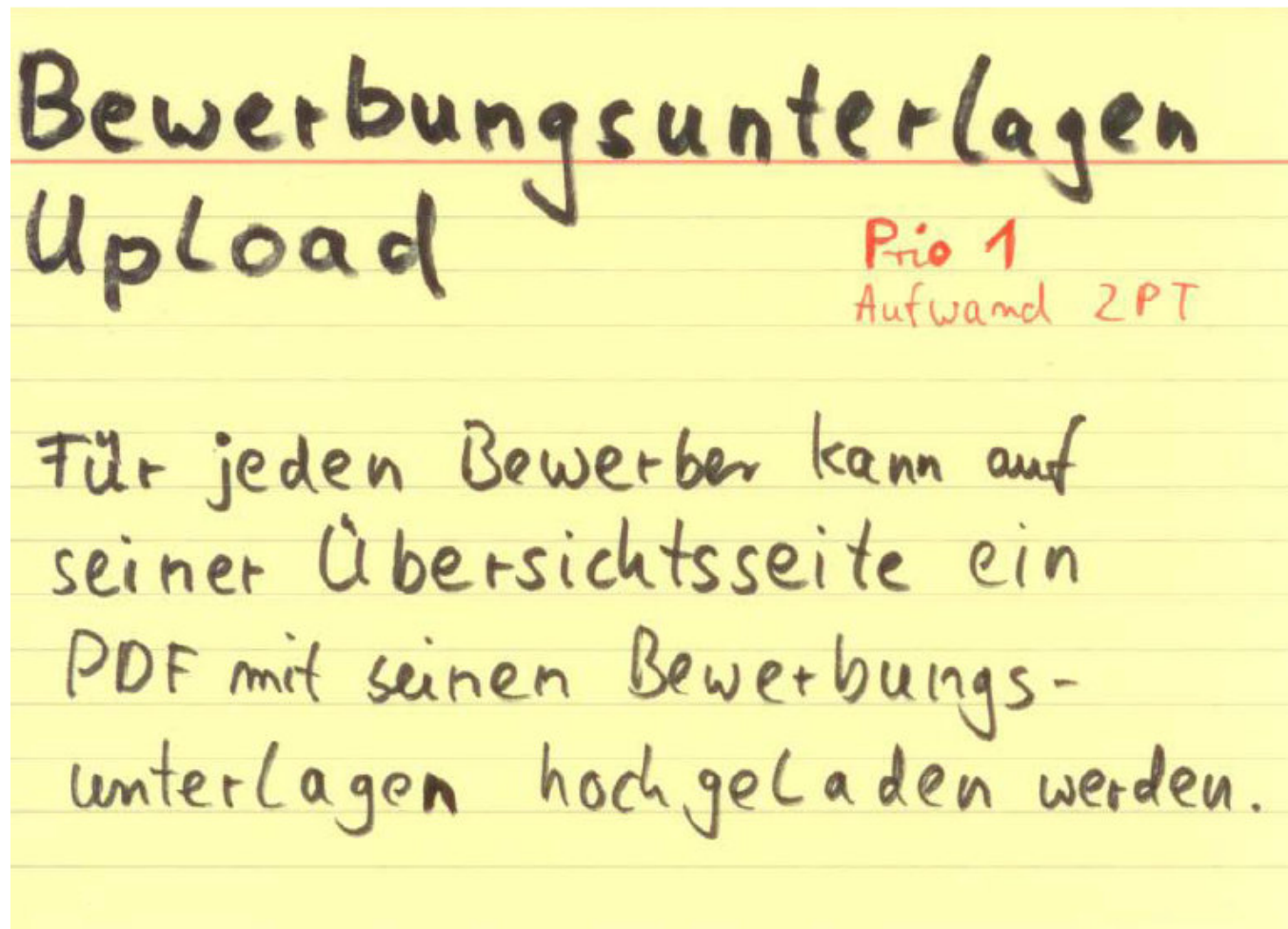
### Beispiel für eine Story-Card



### Beispiel für eine Story-Card



### Story Card



### Beispiele für StoryCards

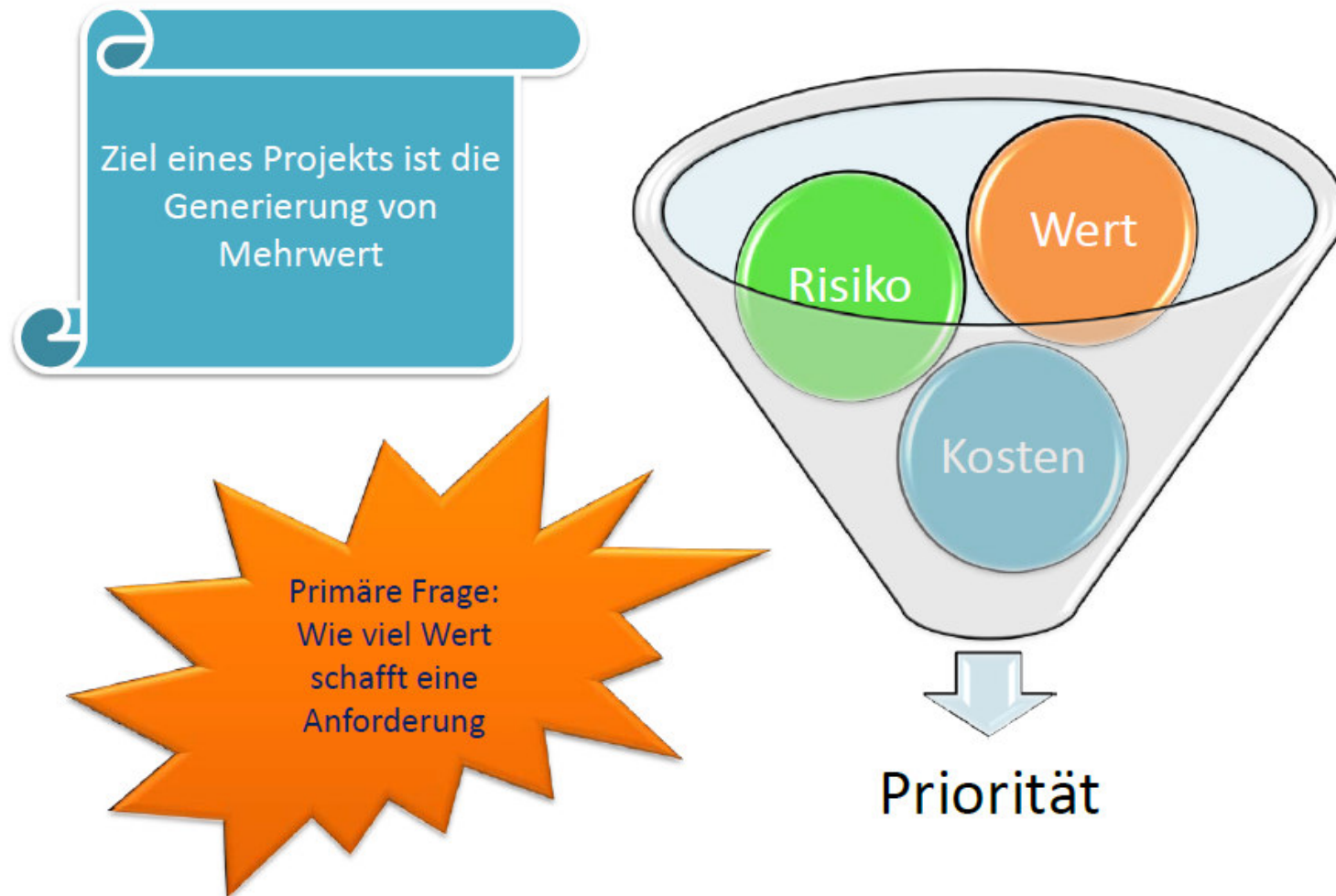
Präsentation
Als Leser
möchte ich die Präsentation mit einem
frei verfügbaren pdf-Reader lesen können
damit ich schnell darauf zugreifen kann

Bewerbungsunterlagen  
Upload

Prio 1  
Aufwand 2 PT

Für jeden Bewerber kann auf  
seiner Übersichtsseite ein  
PDF mit seinen Bewerbungs-  
unterlagen hochgeladen werden.

## Kriterien für eine Priorisierung



### In 5 Schritten zum Product Backlog

1. Eigenschaften, die aus dem Produktkonzept folgen, sowie Anforderungen, die durch Fokusgruppen, Kundeninterviews, Anforderungswshops,... ermittelt wurden, kommen in das .....
2. .... gruppiert Anforderungen zu Themen
3. .... priorisiert die Themen und ggf. einzelne Anforderungen (in Zusammenarbeit mit dem Team)
4. Product Owner verfeinert die .....(vorzugsweise in Anforderungswshops) unter Einbeziehung des Teams
5. Aktualisierung, .....existierenden Anforderungen anhand der Ergebnisse nachfolgender Anforderungswshops. Neue Themen und Anforderungen werden aufgenommen.

R. Pichler: Scrum, dpunkt.Verlag, 2008

### Schätzen im Scrum Prozess

- Im agilen Umfeld löst man sich bewusst von Aufwandsschätzungen in Mann-/Personentagen.  
Warum?
  - Entwickler programmieren .....
  - Schätzungen sind immer ..... und enthalten nicht alle.....
- Geschätzt wird die Komplexität einer Aufgabe.  
Ist etwas ..... als eine andere Aufgabe.
- Stories enthalten Storypoints nach der von Mike Cohn angepassten Fibonacci-Reihe  
1,2,3,5,8,13, 20, 100

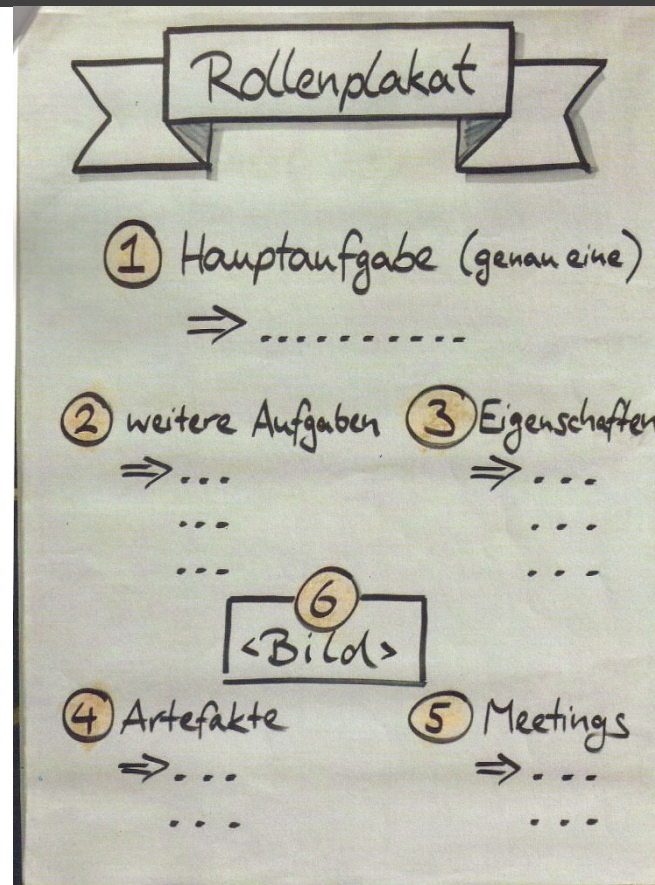


### Akzeptanz-/Testkriterien für User Stories

- Jede User Story enthält ....., die beschreiben wann die Story erfolgreich implementiert wurde (done)
- Es wird beschrieben, .....sind
- Welche wichtigen Tests müssen erfolgreich durchgeführt werden

### Aufgabe zum Kennenlernen der Scrum-Rollen

Bilden Sie sechs Gruppen von denen immer zwei Gruppen sich eine Scrum-Rolle teilen. Erarbeiten Sie die Rolle nach Vorlage des Rollenplakats. Immer beide Gruppen zusammen stellen ein Rollenplakat dar.



# Rollen



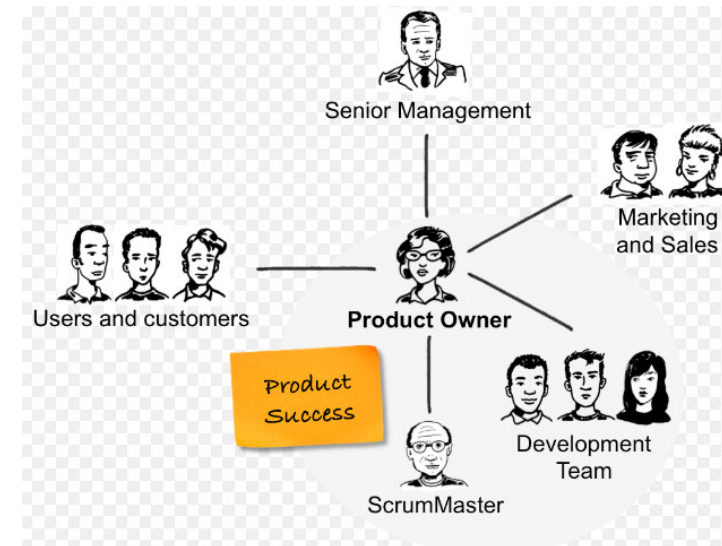
### ...und wo ist der Projektleiter abgeblieben?

Aufgabenbereich	Scrum-Rollen
Scope-Management	<ul style="list-style-type: none"><li>• Product Owner für Produktversion</li><li>• Team für Sprints</li></ul>
Zeitmanagement	<ul style="list-style-type: none"><li>• Product Owner für den Releaseplan</li><li>• Team für das Sprint Backlog</li></ul>
Kostenmanagement	Product Owner
Kommunikationsmanagement	<ul style="list-style-type: none"><li>• Product Owner für das Release-Reporting</li><li>• Team für Berichterstattung im Sprint</li></ul>
Risikomanagement	Product Owner mit Input vom Team
Qualitätsmanagement	<ul style="list-style-type: none"><li>• Product Owner für die Produktleistungsmerkmale</li><li>• Team für qualitätssichernde Maßnahmen</li><li>• Scrum Master für die richtige Anwendung des Prozesses</li></ul>
Lieferantenmanagement	Product Owner und Team
Personalmanagement	<ul style="list-style-type: none"><li>• Management für die Bereitstellung der Mitarbeiter</li><li>• Team für Produktivität und kontinuierliche Prozessverbesserung</li></ul>

# Scrum Rollen

## Der Product-Owner

- ..... das Produkt und die Produkt-Features
- Erstellt die .....
- Erstellt und pflegt das .....  
.....
- Priorisiert das Produkt Backlog und legt fest was in den .....  
..... kommt
- Ist für das ..... verantwortlich
- Akzeptiert oder weist  
..... zurück
- Repräsentiert sämtliche .....



# Scrum Rollen

## Der Scrum-Master

- Ist verantwortlich für die Einhaltung des .....  
.....
- Er schützt das Team vor allen ..... und beseitigt auftretende Hindernisse (den Prozess betreffend)
- Er hält den ..... zwischen Team und ..... in Gang
- Er moderiert .....
- Er strebt maximalen ..... und ständige Optimierung an
- Hat die Aktualität der ..... (Sprint Backlog, Burndown Chart) im Blick



# Scrum Rollen

## Das Team

- Das ideale Team besteht aus ..... Personen
- Es ist ..... aufgestellt (Entwickler, Architekten, Tester usw.)
- Das Team ..... sich selbst
- Es zerlegt eigenständig ..... (Anforderungen) in .....
- Es trifft sich täglich zum .....
- Es zeigt das Ergebnis eines Sprints im Sprint Review und ist zu einer kritischen Rückschau in der ..... bereit





### Der Sprint

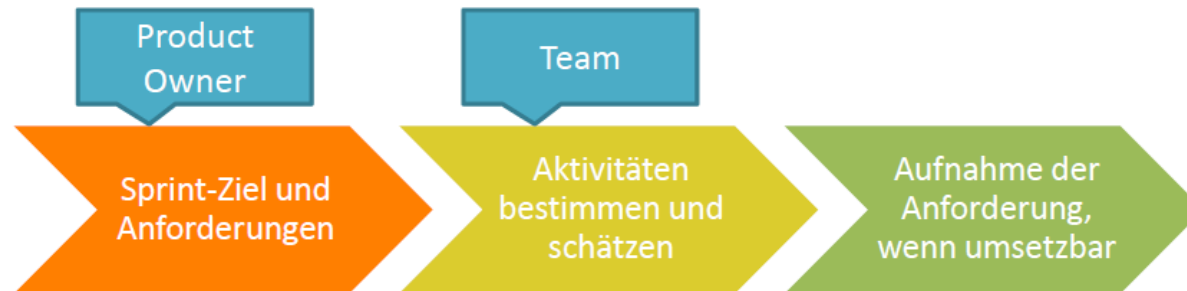


- Hat ideal eine Länge von .....
- Er beginnt mit dem ..... Meeting
- Er endet mit dem ..... und der anschließenden .....
- Jeder Sprint enthält fest definierte Tasks und wird **nicht** durch neue oder geänderte Anforderungen unterbrochen
- Am Ende jedes Sprints steht ein .....



### Sprint-Planungssitzung

- Dient zur Auswahl eines realistischen Sprint Backlog
- Scrum-Master moderiert



Teamkapazität muss im Auge behalten werden

### Typische Fehler bei der Sprintplanung

- Scrum-Master plant
- Ein Teammitglied dominiert
- Viel Design-Diskussion
- ..... identifiziert Aktivitäten
- Product-Owner nimmt .....
- Aktivitäten zu ..... oder zu .....
- Sitzung .....vorbereitet

### Daily Scrum

- Max. 15 Minuten
- Immer zur gleichen Zeit am selben Ort
- Ziel: Selbstorganisation des Teams unterstützen, Hindernisse identifizieren
- Wer: Das komplette Team, Scrum-Master. Product-Owner sollte (wenn möglich) dabei sein, weitere Interessenvertreter als Zuhörer möglich



#### Drei Fragen für jeden:

- Was habe ich seit dem .....erreicht ?
  - Was werde ich bis zum nächsten Daily Scrum erreichen?
  - Was behindert .....?
- 
- Hindernisse werden mit Datum versehen notiert
  - Zur Vorbereitung eines effizienten Daily Scrum muss das aktuelle Sprint Backlog und Sprint Burndown-Chart vorliegen

# Sprint Review

**Ziel:** Begutachtung der Arbeitsergebnisse

- Product-Owner prüft, ob alle akzeptierten Anforderungen vollständig und fehlerfrei umgesetzt wurden
- Dauer ca. 1 bis 2 Stunden
- Wer: Komplettes Team, Product-Owner, Scrum-Master, optional weitere Interessenvertreter (Kunde, Endanwender...)

### Ablauf:

1. Kurze Reflektion über das .....
2. Team macht Live-Demo der Umsetzung .....
3. Product-Owner und Interessenvertreter stellen ..... und geben .....
4. Product-Owner entscheidet für die ..... ob akzeptiert oder nicht

## The Sprint Review



# Sprint Retrospektive

**Ziel:** Zusammenarbeit im Team und Anwendung des Prozesses verbessern

- Wann: Direkt nach dem Sprint-Review
- Dauer: 1,5 bis 2,5 Stunden
- Wer: .....  
..... Nach Bedarf werden weitere  
Interessenvertreter (z.B. Führungskräfte)  
eingeladen
- .....

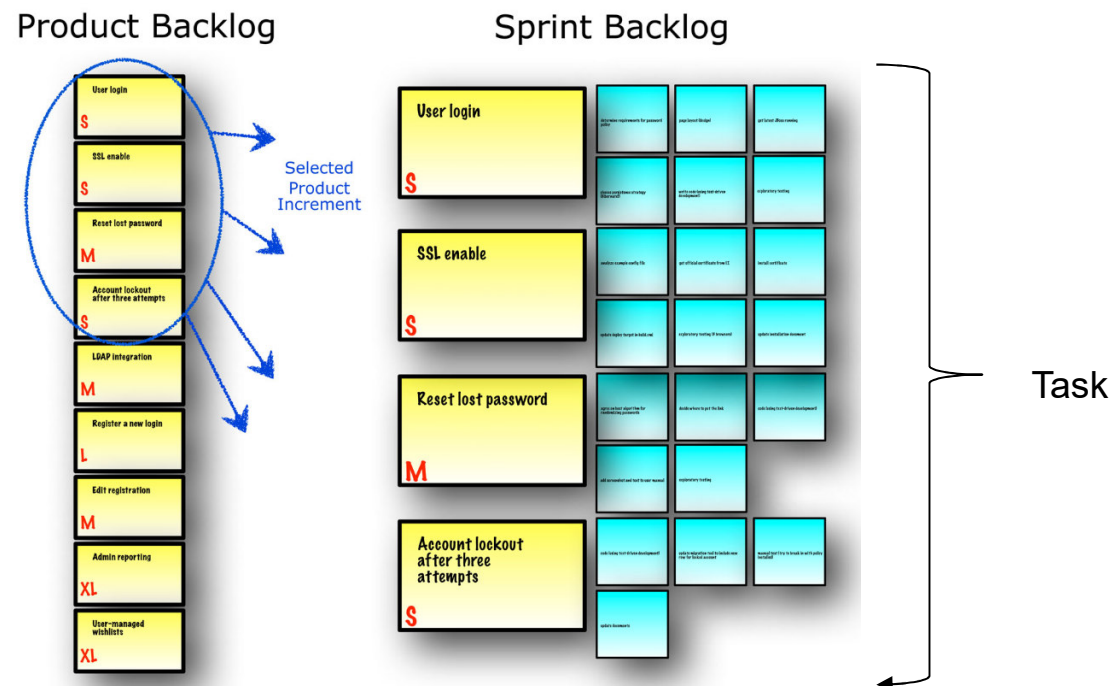
### Ablauf:

1. Daten .....
2. Erkenntnisse .....
3. Entscheidungen .....
4. ....



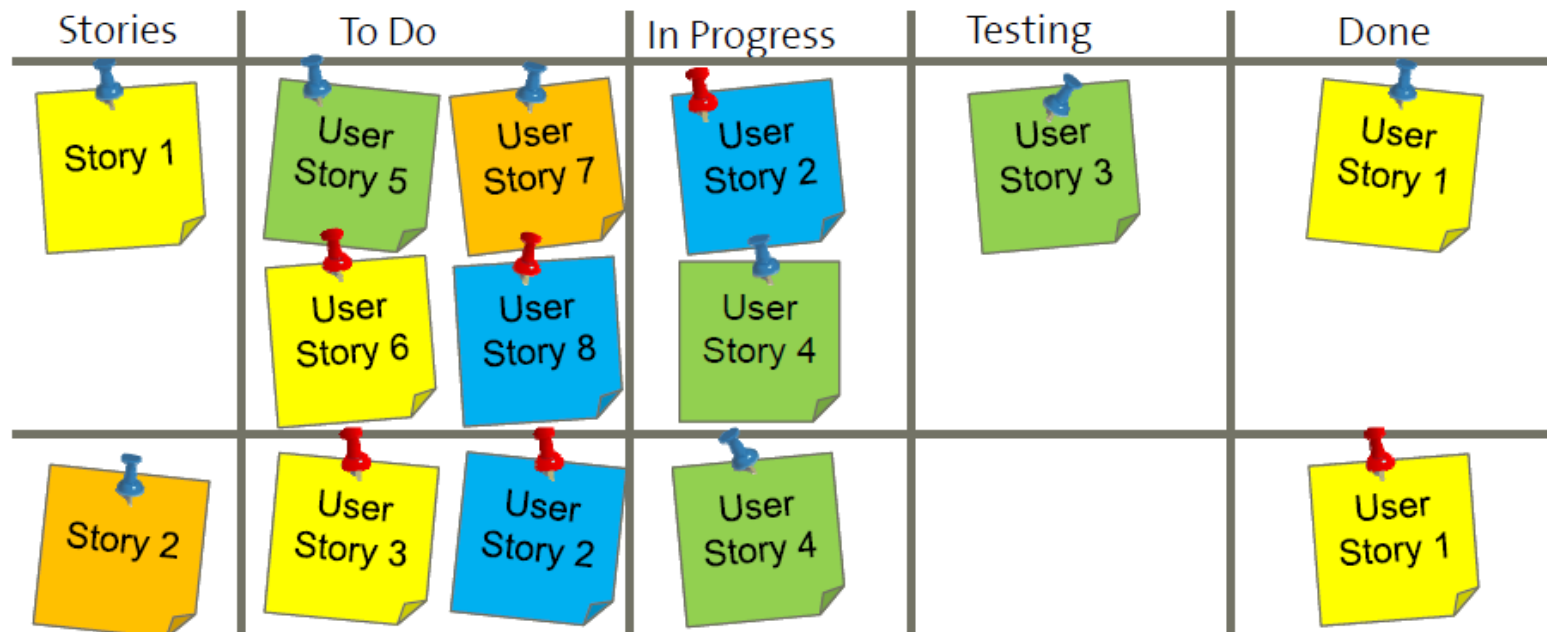
# Vom Product Backlog zum Sprint Backlog

1. Team wählt User Stories/Tasks(nicht functional) für den nächsten Sprint
2. User Stories werden in einzelne Tasks .....
3. Tasks in ..... schätzen

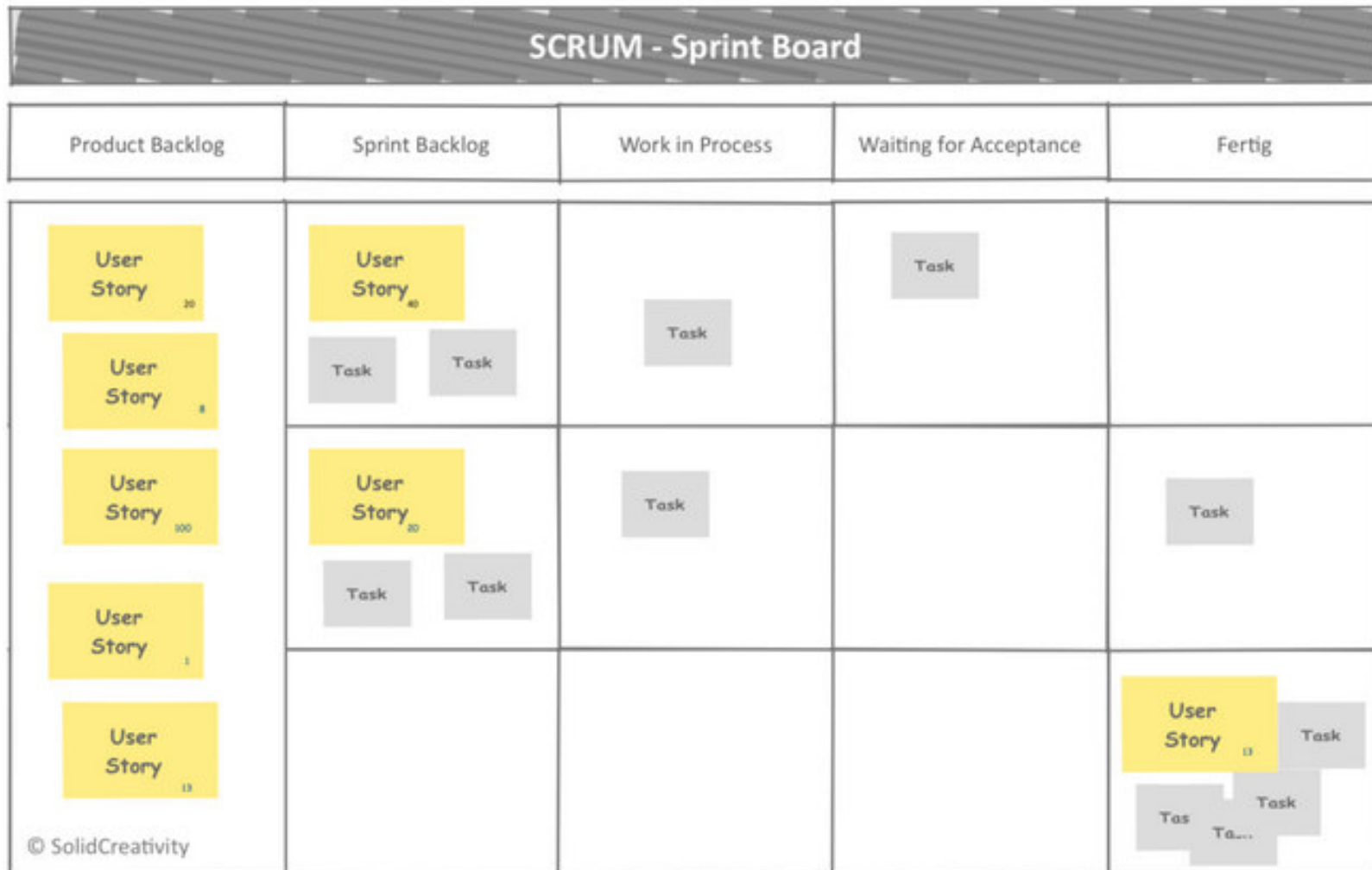


# Scrum - Informieren und Steuern

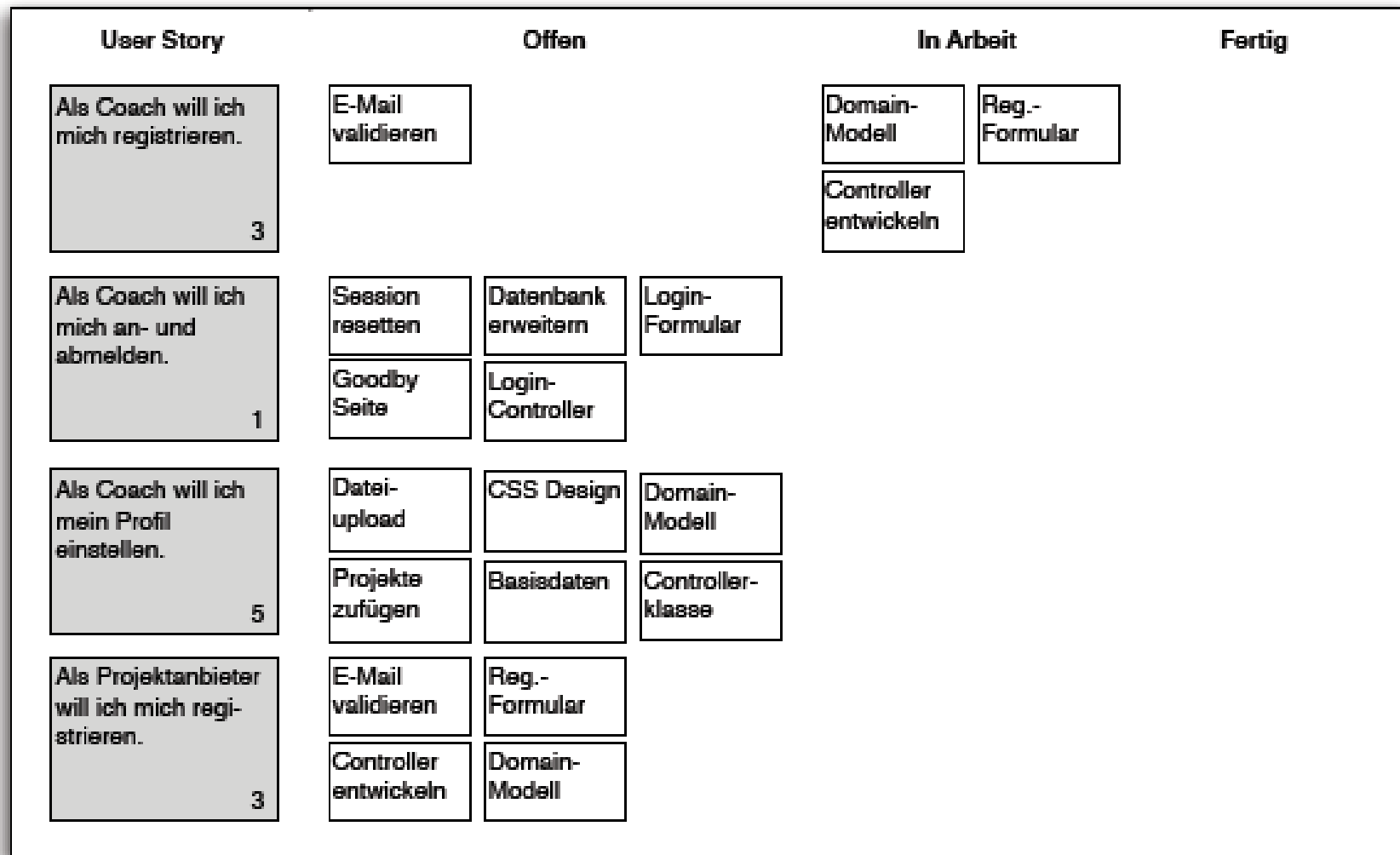
## Scrum Task Board



## 11 – Agiles Vorgehen und SCRUM



## 11 – Agiles Vorgehen und SCRUM



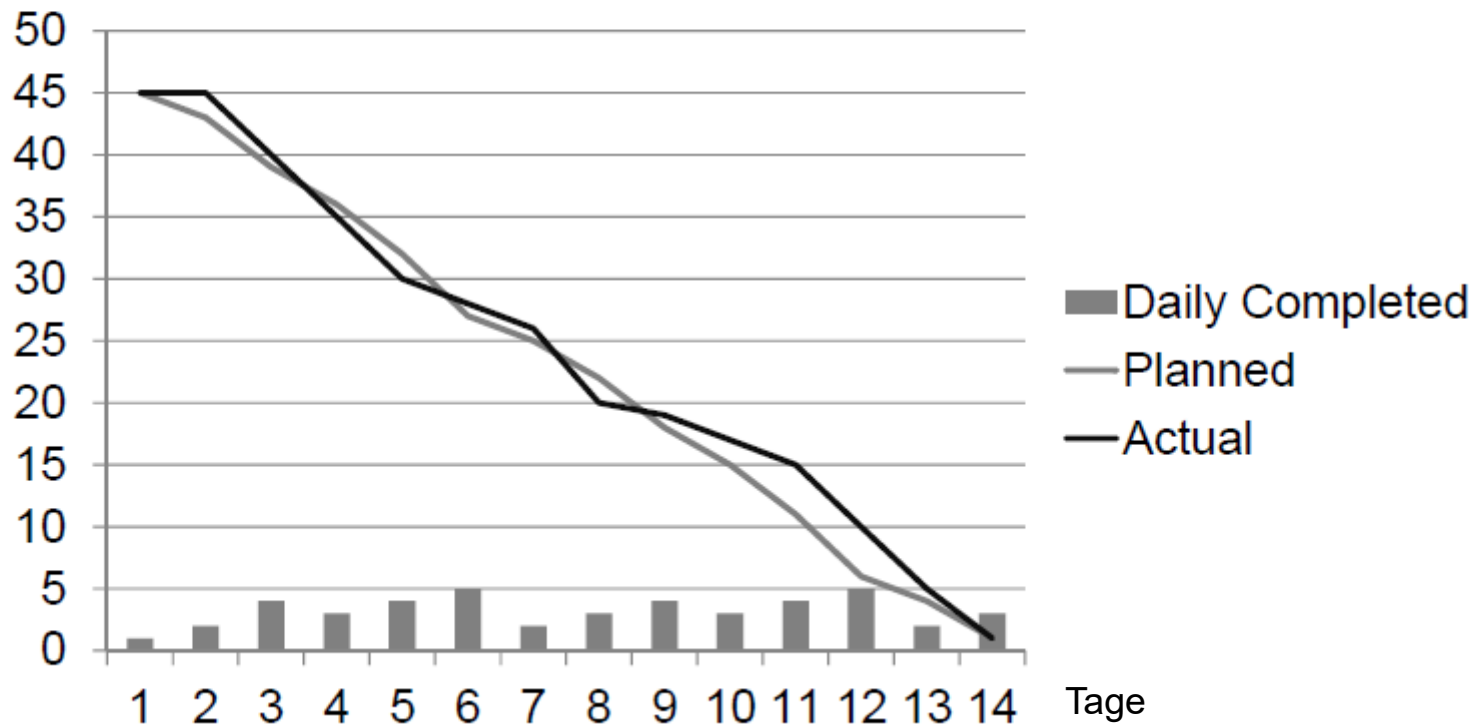
([1], S.163)



# Scrum - Informieren und Steuern

## Burndown Chart

Story-Points



### Fazit & Ausblick

#### Erfolgsversprechender Ansatz.

- praxiserprobt
- ergebnisorientiert
- schnelle erste lauffähige Inkremente
- Team übernimmt Verantwortung
- Vorgehen fördert „agile Werte“



#### Voraussetzungen.

- Vertrauen
- Projektstruktur und -größe
- durchgängige Kommunikation
- festes Team mit allen nötigen Kompetenzen
- konsequente Umsetzung



## Agiles Vorgehen und SCRUM



Ende des Kapitels

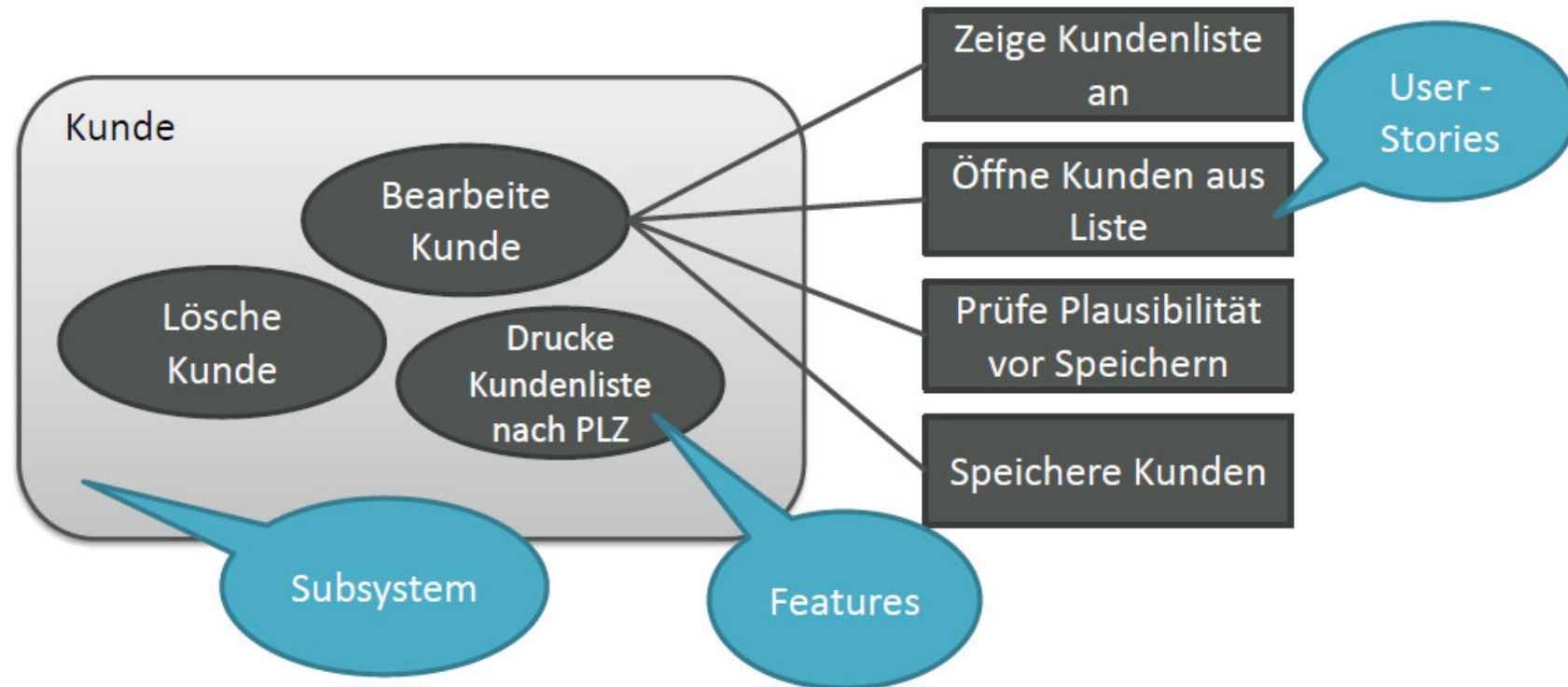
# Fragen



## Backup-Folien



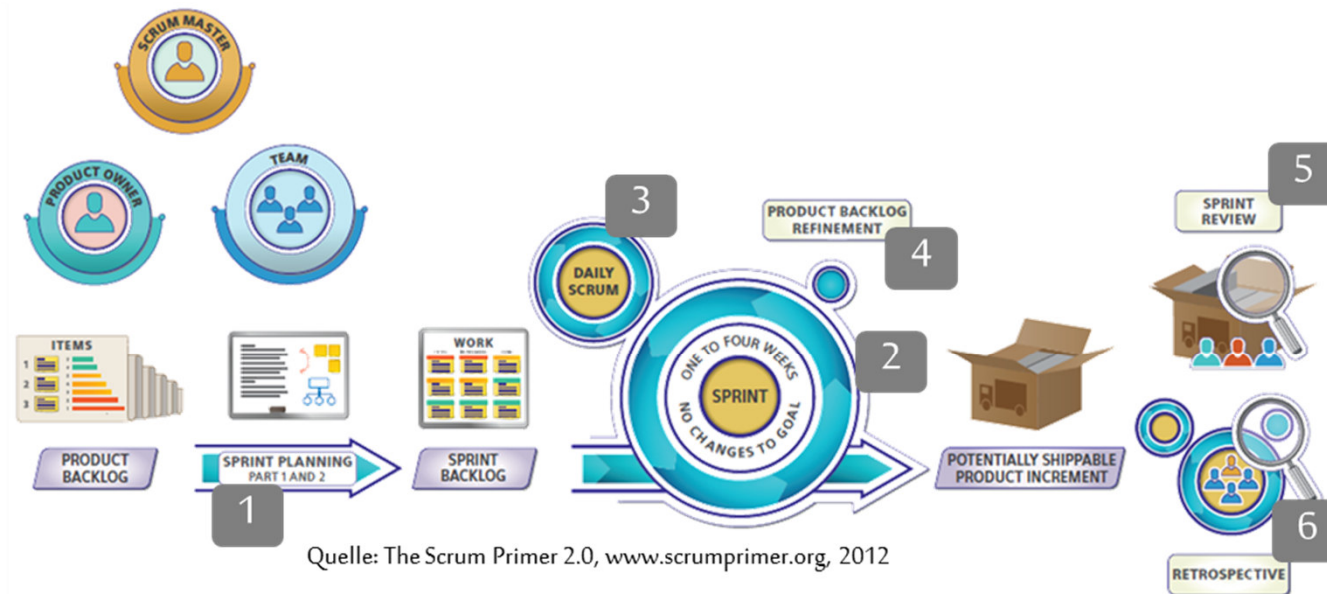
## Hierarchische Schätzverfahren



### Entwicklungsphase

- Am Anfang: Entwickler sprechen sich ab, wie in der Iteration vorzugehen ist und wie das grobe Design für die Funktionalitäten auf den Story Cards aussehen soll.
- Implementieren der Funktionalitäten einzelner Story Cards (im paarweisen Programmieren).
- Tracker beobachtet Fortschritt der Implementierung (um im nächsten Planning Game beim Schätzen der Aufwände zu helfen).
- Kommt es in der Iteration zu Unklarheiten, werden diese mit den anderen Entwicklern oder mit dem Product Owner geklärt.
- Unit Tests sollte für geringe Fehlerquote sorgen. Akzeptanztests werden ausgeführt.
- Der Abschluss einer Iteration, nach der zuvor festgelegten Dauer (1 bis 4 Wochen), ist eine funktionierende Software.

### Aktivitäten



#### **Sprint Planning:**

- Teil 1: Sprint-Umfang festlegen
- Teil 2: Sprint-Verlauf planen

#### **Daily Scrum:**

- Tägliches Statusmeeting (15min)

#### **Sprint Review:**

- Begutachtung d. Sprint-Ergebnisses
- Planung des weiteren Vorgehens

#### **Sprint:**

- Fester Zeitraum (2 – 4 Wochen)
- Ziel: Lieferfähiges Inkrement

#### **Product Backlog Refinement:**

- Ständiges anpassen und verfeinern
- Team unterstützt Product-Owner

#### **Retrospective:**

- Analyse des letzten Sprints
- Verbesserungen ableiten u. planen



## Wesentliche Ergebnisse



### Literatur

**[1] Wirdemann, Mainusch:** SCRUM mit User Stories  
3. Auflage, Hanser-Verlag 2017

**[2] Boris Gloger:** SCRUM Produkte zuverlässig und schnell entwickeln  
5. Auflage, Hanser-Verlag 2016

**[3] Karl-Heinz Rau:** Agile objektorientierte Software-Entwicklung  
1. Auflage, Springer Vieweg Verlag 2015

**[4] Dräther, Koschek, Sahling:** SCRUM kurz & gut  
1. Auflag, O'Reilly Verlag 2013