



# Professionelles Projektmanagement in der Praxis - mit digitalen Unternehmensgründungsprojekten

Prof. Dr. Harald Wehnes

Veranstaltung 4 (11.05.2020):  
**Vorgehensmodelle im Projektmanagement**  
**Scrum2: Wiederholung und Ergänzungen**

**Partner:**

Servicezentrum  
Forschung und  
Technologietransfer  
(SFT)

  
Innovations- & Gründerzentrum  
WÜRZBURG



# Agenda

---

- ▶ Organisatorisches
- ▶ **Vorgehensmodelle**
- ▶ Aufgabe 3: Ergebnisse der Projektleiter der Teams 1 - 6
- ▶ **Scrum (Wiederholung + Ergänzungen)**
  - *Workshop: User Stories (MVP) → Tasks*
  - Aufgabe 4
- ▶ **Zertifizierungen im Projektmanagement**

**Gastvortrag Frau Golly (SFT) „Business Planning“ am 15. Mai, 10:00 Uhr**

# Aktualisierter Terminplan

Tag	Zeitraum	Inhalt	Anmerkungen
<b>Montags, ab 20.04.2020 bis einschließlich 20.07.2020</b>	<b>12:15 - 15:45 Uhr</b>	<b>Vorlesung</b>	<b>Einwahl um 12:00 Uhr, um pünktlich zu starten</b>
15.04.2020	09:30 - 16:30 Uhr	Design Thinking Workshop	Einweisung der WS- Teilnehmer am 14.04.2020
17.04.2020	11:00 - 12:00 Uhr	Technischer Test	für alle Vorlesungsteilnehmer
<b>15.05.2020</b>	<b>10:00 Uhr: Gastvortrag Frau Golly (SFT) „Business Planning“</b>		
<b>01.06.2020</b>	<b>Keine Vorlesung</b>		
<b>20.07.2020*</b>	<b>12:30 - 16:00 Uhr</b>	<b>Projektiade 2020</b>	<b>Öffentliche Abschlussveranstaltung</b>
<b>27.07.2020*</b>	<b>12:00 - 13:20 Uhr</b>	<b>Klausur zur Vorlesung</b>	Dauer: 80 Minuten
<b>27.07.2020*</b>	<b>16:00 - 17:00 Uhr</b>	<b>Klausur: Hochschul- Zertifikat "Modernes PM"</b>	<b>Optionale Teilnahme</b> Dauer: 60 Minuten

\*) Änderungen möglich

Stand: 08.05.2020

# Terminplan der Vorträge

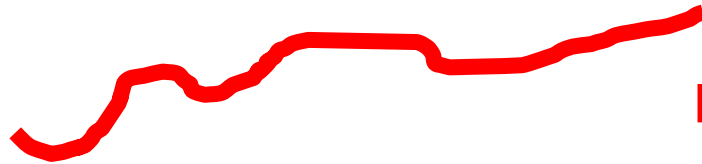
Team	Anzahl Mitglieder	04.05.	11.05. + 15.05.	18.05.	25.05.	08.06.	15.06.	22.06.	29.06.	06.07.	13.07.	20.07.
1	5		<b>x</b>		x		x		x			XX
2	7		<b>x</b>	x	x		x		x		x	XX
3	6		<b>x</b>		x		x		x		x	XX
4	6	x	<b>x</b>			x		x		x		XX
5	7	x	<b>x</b>	x		x		x		x		XX
6	5	x	<b>x</b>			x				x		XX

---

# VORGEHENSMODELLE

# Vorgehensmodelle im Projektmanagement

**Definition:** Vorgehensmodelle sind systematische und koordinierte Vorgehensweisen für die Abwicklung eines Projektes



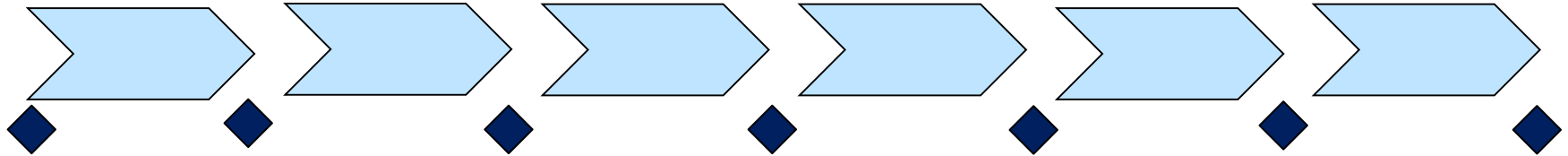
Der rote Faden durch den Projektlebensweg

## Nutzen

- Standardisierung
- Klarheit und Transparenz beim Vorgehen
- Erleichterung der Projektarbeit
- höhere Qualität der Projektergebnisse
- „Best Practices-Vorgehensweise“ für bestimmte Branchen etc.

# Bausteine von Vorgehensmodellen

- ▶ **Phasenmodell** mit Meilensteinen und Meilensteinergebnissen



- ▶ **Projektprozesse (Aktivitäten)**

- ▶ **Methoden und Werkzeuge**

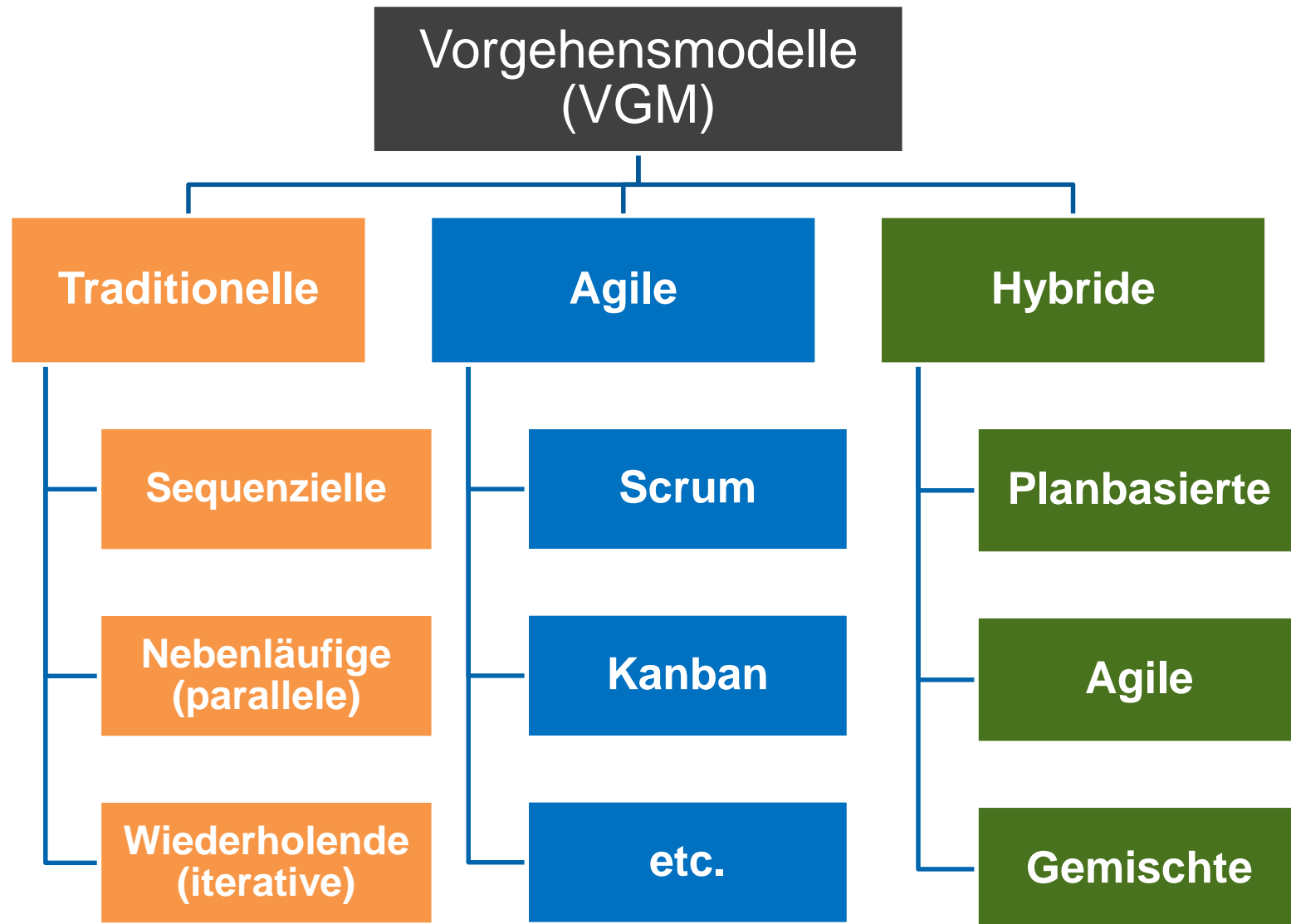
- ▶ **Dokumente**

(wie z.B. Checklisten, Dokumentationsvorlagen, Pläne, Templates)

- ▶ **Rollenkonzept:** Rollen und Rollenbeschreibungen

*Nicht jedes Vorgehensmodell muss notwendigerweise alle Bausteine enthalten*

# Vorgehensmodelle im Überblick





# Traditionelle Vorgehensmodelle

**Gemeinsame Eigenschaft traditioneller VGM: Es wird ein Plan erstellt.**  
**Mit der korrekten Umsetzung des Plans wird das Projektziel erreicht**

## ▶ **Sequenzielle** Vorgehensmodelle

- Projekt wird in Phasen strukturiert, die nacheinander ablaufen
- *Vertreter: Wasserfallmodell, Stage-Gate-Modell, V-Modell, V-Modell XT*

## ▶ **Nebenläufige (parallele)** Vorgehensmodelle

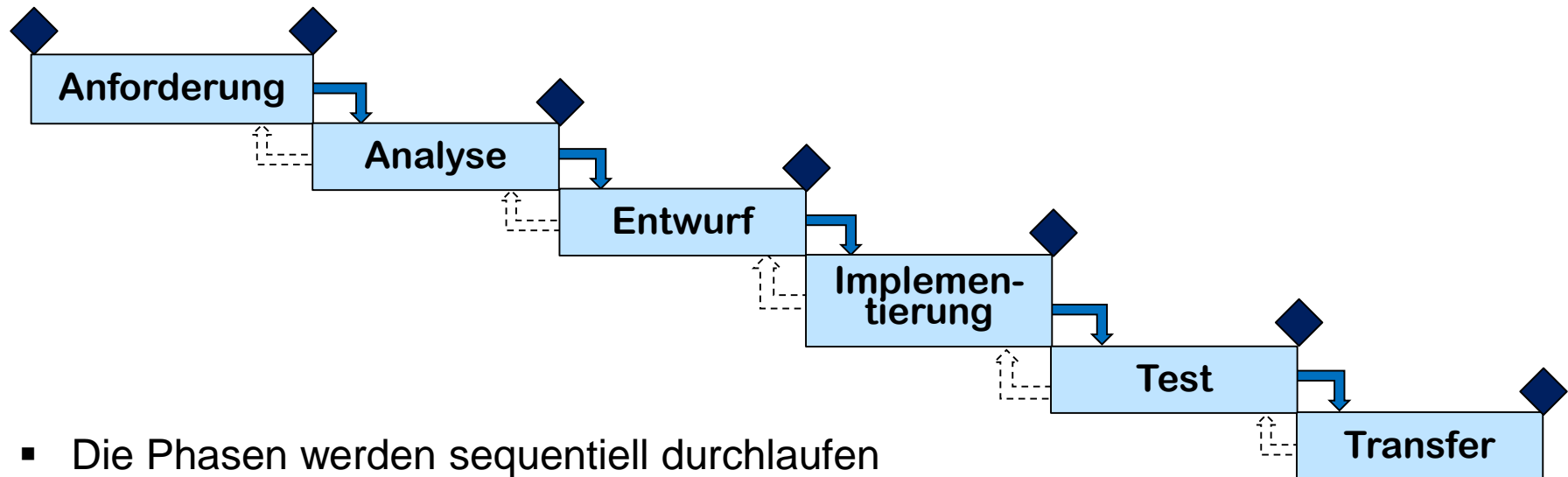
- Projektphasen überlappen sich, um durch parallele Bearbeitung Zeit zu sparen
- *Vertreter: Simultaneous Engineering*


## ▶ **Wiederholende (iterative)** Vorgehensmodelle

- Phasen werden mehrfach durchlaufen; Projektergebnis wird schrittweise erarbeitet
- *Vertreter: Inkrementelles Vorgehensmodell, Spiralmodell*

# Sequenzielle VGM: Wasserfallmodell

Streng lineare Vorgehensweise mit in sich abgeschlossenen Phasen  
Jede Projektphase hat einen Start- und Endpunkt und definierte Ergebnisse



- Die Phasen werden sequentiell durchlaufen
- Rücksprünge in die unmittelbar zurückliegende Phase sind möglich, z.B. bei Projektänderungen
- **Meilensteine**  am Phasenende
- **Qualitätssicherung** durch geordnete Phasenübergänge: Liegen alle Ergebnisse vor?

## Stärken

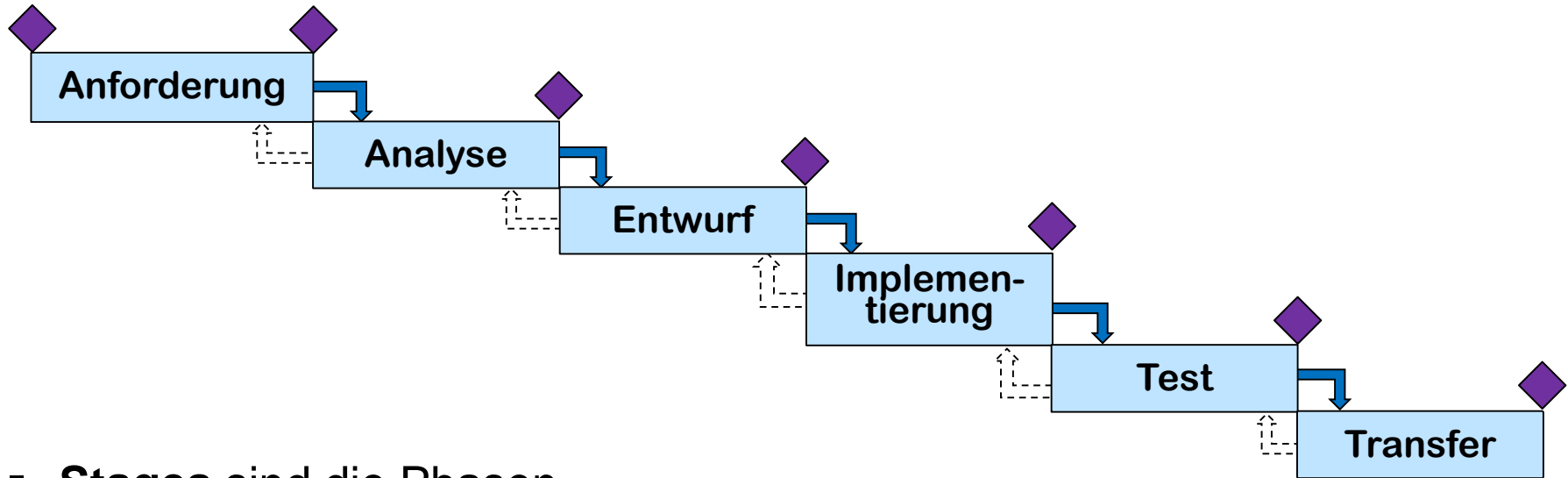
- + sehr einfaches Modell
- + wenig Management-Aufwand
- + nachvollziehbare Planung und Kontrolle

## Schwächen

- ineffizient bei sich ändernden oder unvollständigen Anforderungen
- ungeeignet für Projekte mit hohen Risiken

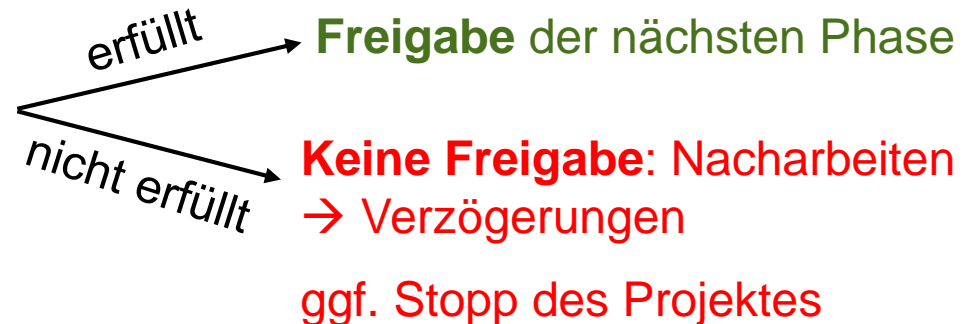
# Sequenzielle VGM: Stage-Gate Modell (Quality-Gate-Modell)

Wasserfallmodell mit erhöhten qualitätssichernden Elementen ◆ Quality Gates



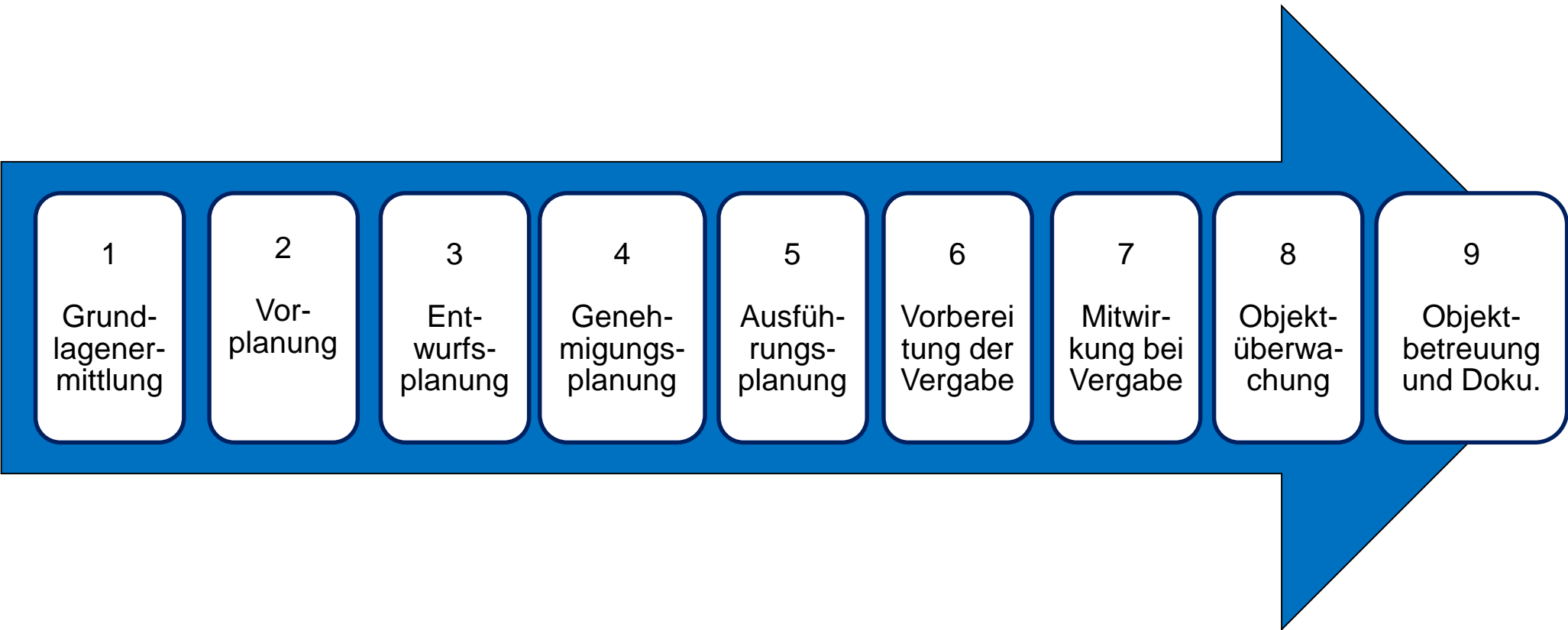
- **Stages** sind die Phasen
- **Gates** ◆ sind die Tore, die am Phasenende zu durchschreiten sind  
**Ist alles in ausreichender Qualität da?**

Gremium prüft an den Gates, ob definierte Kriterien, z.B. Vorliegen von Ergebnissen (Zwischenprodukte, Dokumente u.ä.) in einer bestimmten Qualität vorliegen



# Sequenzielle VGM: Phasenmodell der Baubranche

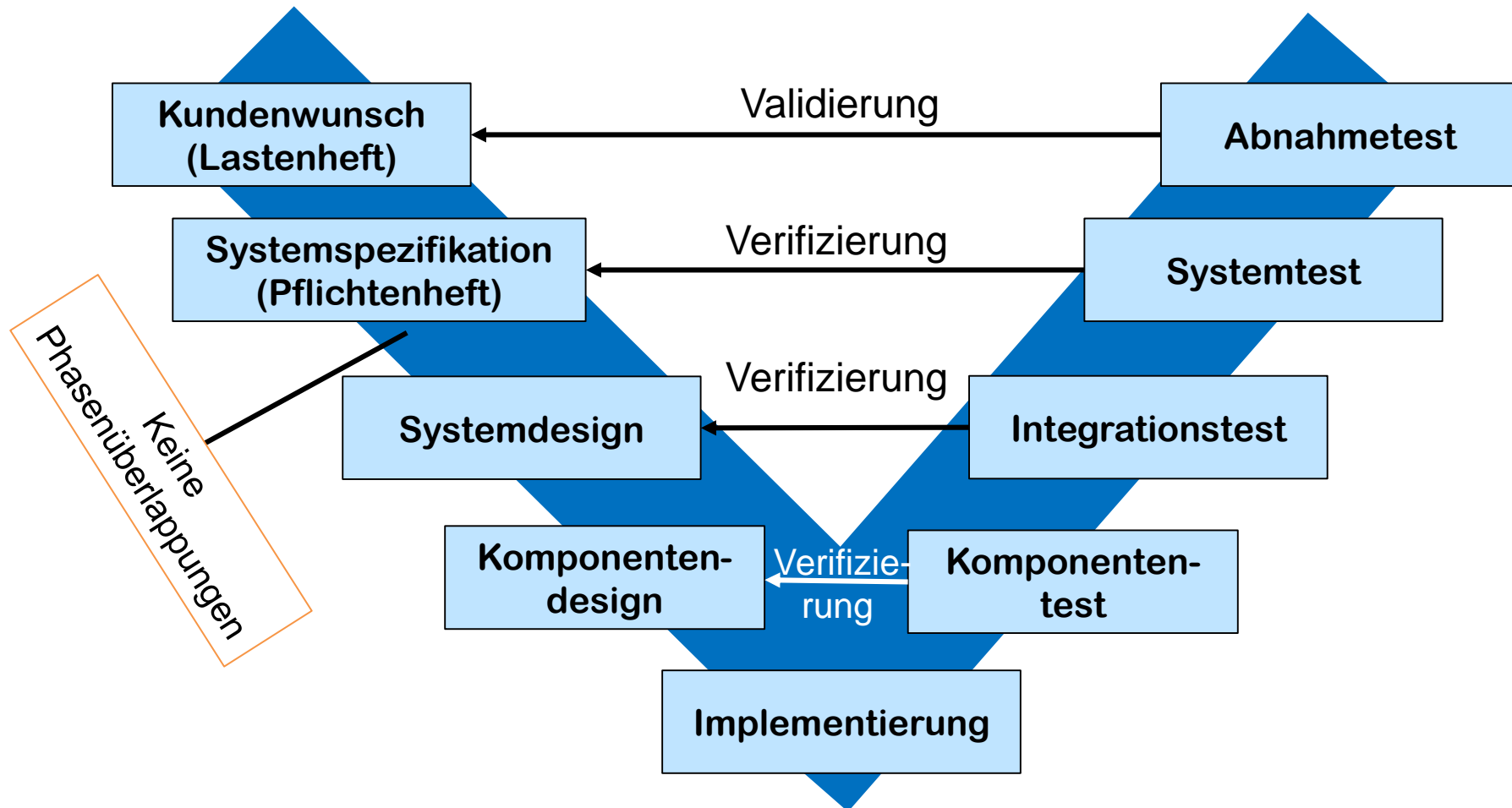
HOAI (Honorarordnung für Architekten und Ingenieure): 9 Leistungsphasen



Der Abschluss der Leistungsphasen stellen natürliche Quality Gates dar

# Sequenzielle VGM: V-Modell

Weiterentwicklung des Wasserfallmodells mit verstärktem Fokus auf **Qualität**



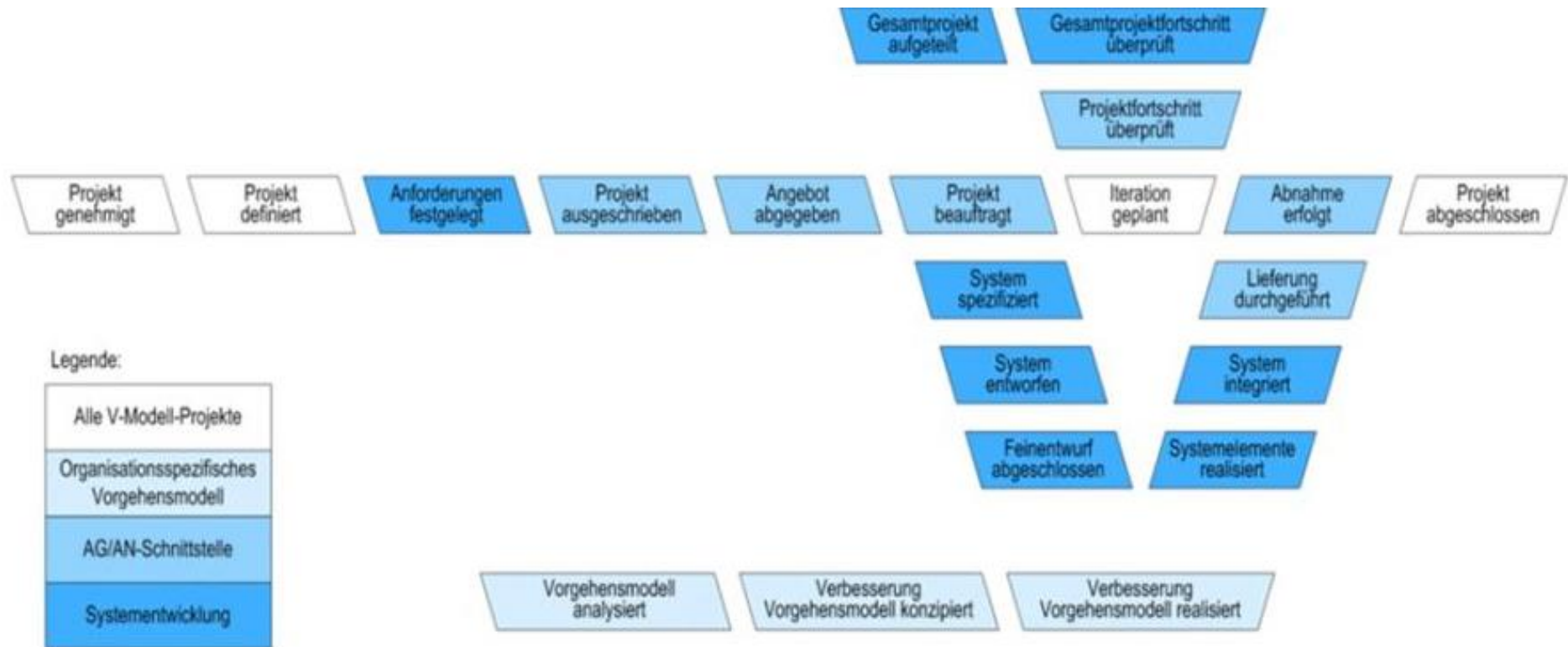
+ sehr gut nachvollziehbare Planung und Kontrolle  
+ hohe Qualität der Dokumentation  
Einsatz für Branchen mit hohem Sicherheitsbedürfnis

- sehr hoher Aufwand bei Änderung von Anforderungen  
- zu aufwändig für kleine Projekte

# Sequenzielle VGM: V-Modell XT



Das V-Modell XT (Vorgehensmodell “Extreme Tailoring”) ist der etablierte und häufig geforderte Standard für die System- und Softwareentwicklung in der Öffentlichen Verwaltung in Deutschland



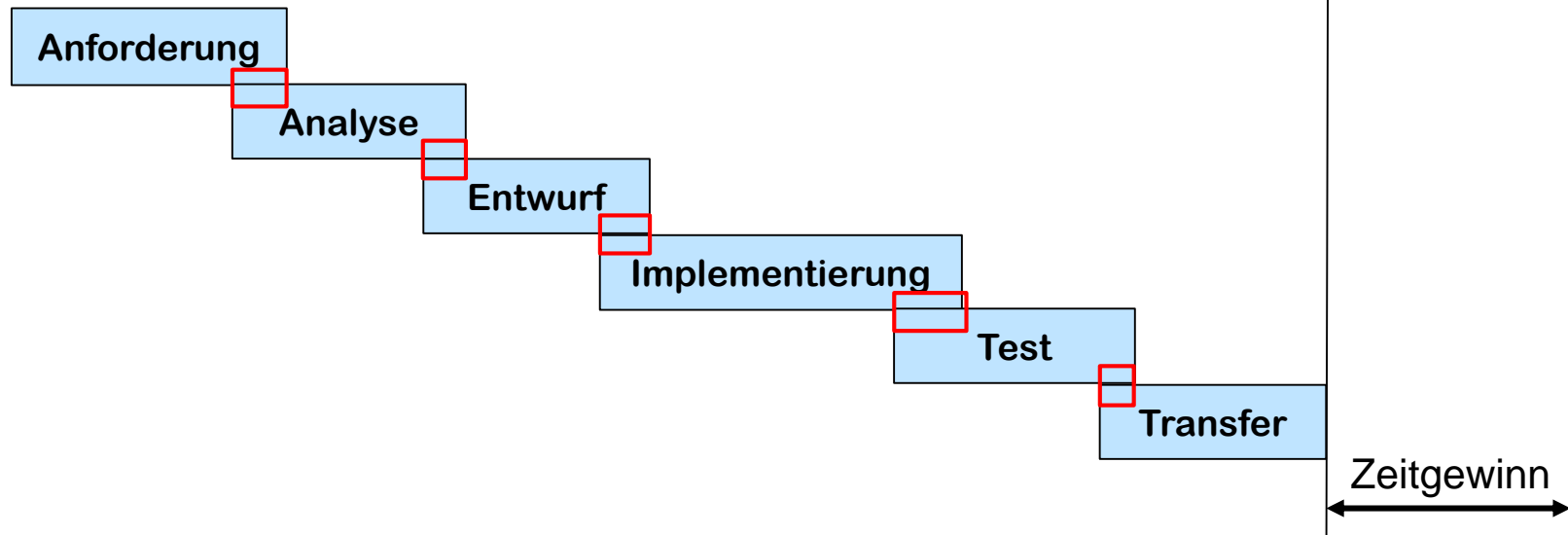
Beschreibung (500 Seiten): <http://ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/Releases/2.0/V-Modell-XT-Gesamt.pdf>

# Nebenläufige VGM: Simultaneous Engineering:

Reine sequentielle Reihenfolge



Simultaneous Engineering: Teil-parallele Reihenfolge



## Stärken

- + Zeitgewinn durch Parallelisierung

## Schwächen

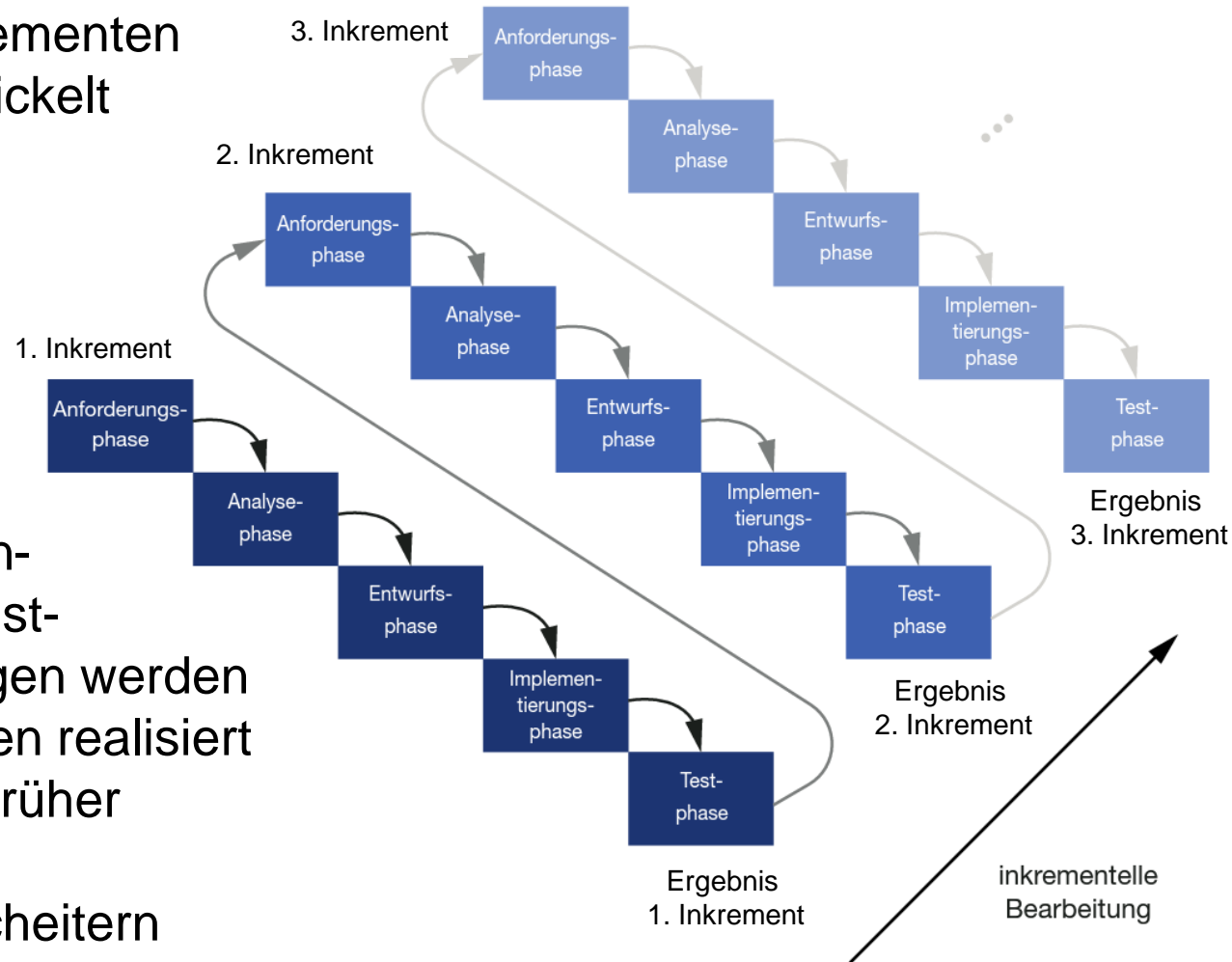
- Risiko von Mehraufwand bei späteren Änderungen
- Fachbereichs-übergreifende Zusammenarbeit erforderlich

# Wiederholende VGM: Inkrementelles VGM

Projektprodukt wird in Inkrementen (Software-Versionen) entwickelt

**Inkrement** := Teilmenge von Anforderungen, die realisiert werden

- Priorisierung der Kundenanforderungen; die höchstpriorisierten Anforderungen werden in den ersten Inkrementen realisiert
- Systemfunktionalität ist früher verfügbar
- Geringeres Risiko für Scheitern des Gesamtprojekts



Quelle: [1], S. 44



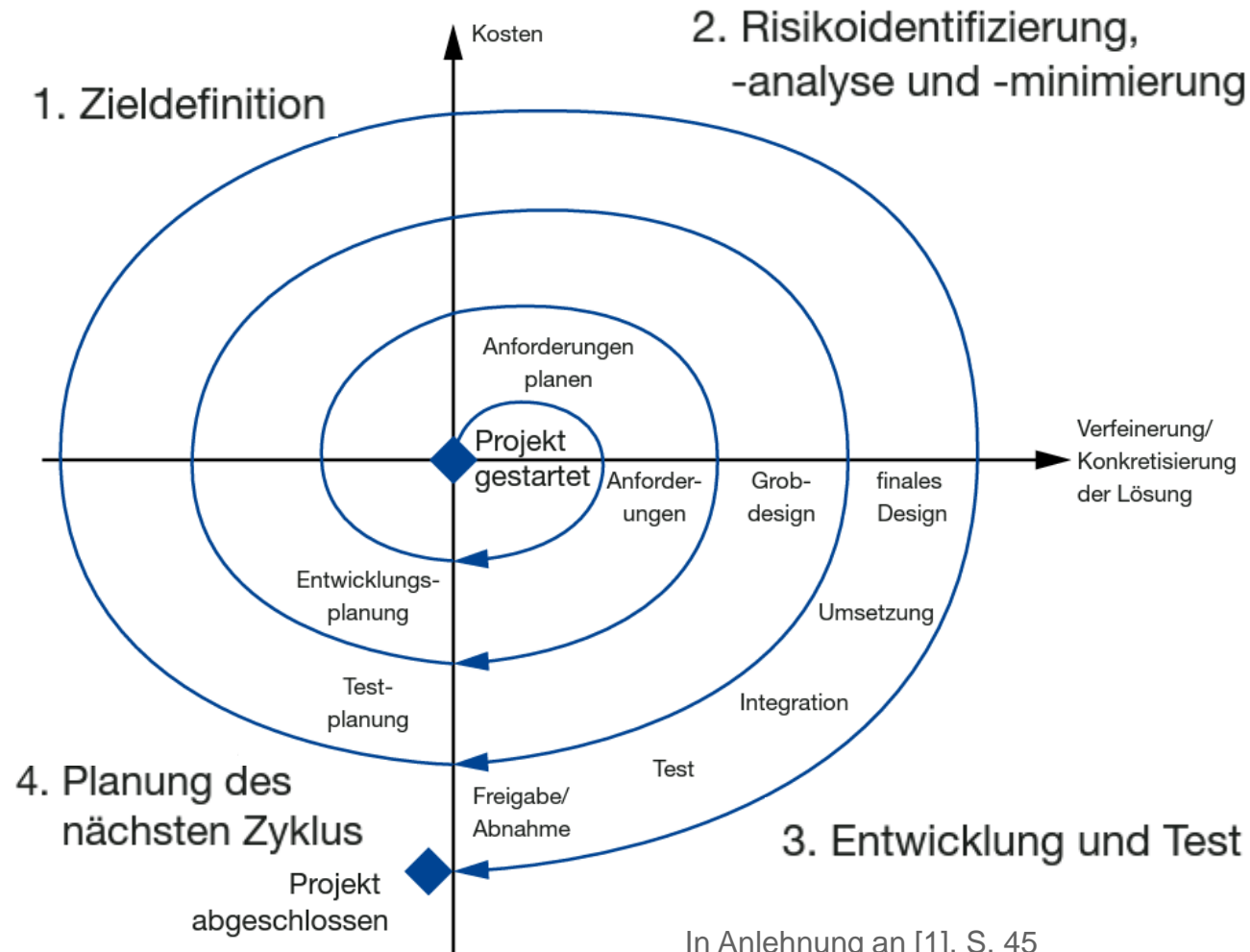
# Wiederholende VGM: Spiral-Modell (nach Boehm)

## Bekanntestes wiederholendes Vorgehensmodell

- Jede Spirale stellt einen iterativen Zyklus durch dieselben Schritte dar
- Die Ziele für jeden Zyklus werden aus den Ergebnissen des letzten Zyklus abgeleitet

+ strukturierter Umgang mit unscharfen Anforderungen  
+ schrittweise Detaillierung des Projektgegenstandes

- Risiko des Mehraufwands bei späteren Änderungen
- ungeeignet für kleine und mittlere Projekte



In Anlehnung an [1], S. 45

# Agile Vorgehensmodelle

## Agile Werte und Prinzipien

### Agiles Manifest:

4 Werte

12 Prinzipien

## Agile VGM (Methoden)

**Scrum,**  
Kanban,  
XP, DevOps  
Lean PM,  
**Lean Startup,**  
**Design Thinking**

## Agile Praktiken

**Product Backlog,**  
**Story Points,**  
**Planning Poker,**  
**Time Boxing,**  
Kanban Board,  
**Daily, Sprints,**  
Selbstorganisation,  
WiP, ...

# Agiles Vorgehen im Vergleich zu traditionellem

## Traditionell

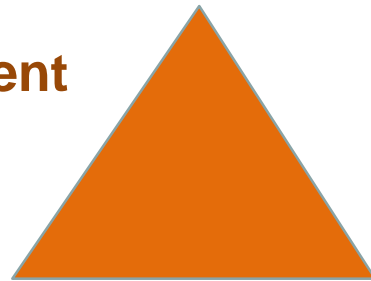


## Agil

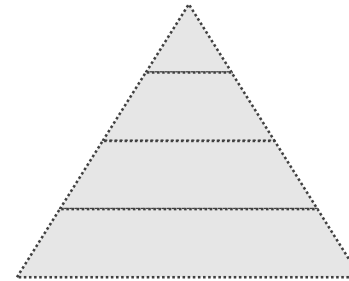


# Prinzip der kleinen Pyramide

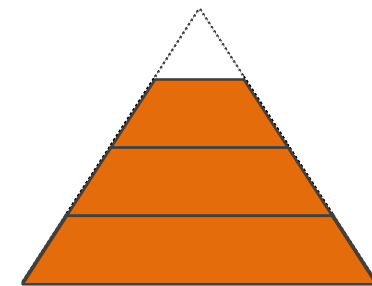
## Traditionelles Projektmanagement



Ziel

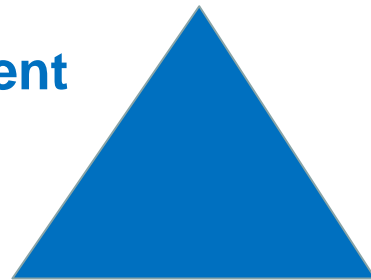


Plan

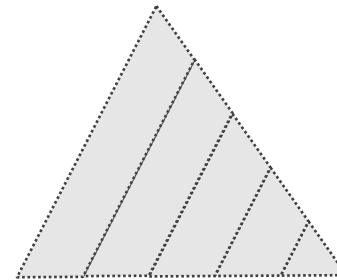


Ergebnis

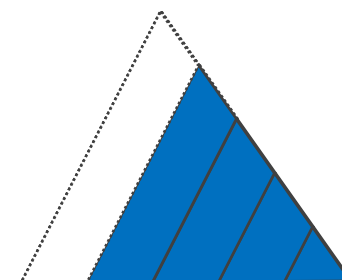
## Agiles Projektmanagement



Ziel



Plan



Ergebnis

Quelle: Philipp Meyerbröker: Das Prinzip der kleinen Pyramide  
Projekt Magazin 10/2010

# Ausblick: VGM II

---

- ▶ Weitere agile VGM
  - Kanban
  - Lean Projektmanagement
  - DevOps
  - XP (Extreme Programming)
  - Design Thinking
- ▶ Modelle zur Auswahl eines Vorgehensmodells
  - Stacey-Matrix
  - Boehm und Turner
  - Cynefin-Framework
- ▶ Tailoring (maßgeschneiderte Vorgehensmodelle konfigurieren)
- ▶ Hybride Vorgehensmodelle
  - Planbasierte hybride VGM (Beispiel: integrierte Anwendung aus Stage-Gate-Modell und V-Modell.
  - Agile hybride VGM (Beispiel: ScrumBan)
  - Gemischte hybride VGM (Beispiel: Water-Scrum-Fall)

# Aufgabe 3: Social Innovators Challenge

Sie wollen mit Ihrem Team am „Social Innovators Challenge“  
<https://www.uni-wuerzburg.de/sft/gruendungsberatung/veranstaltungen/social-innovators-challenge/> (bzw. einen anderen Startup-Wettbewerb) teilnehmen.

1. Erstellen Sie ein **Ideenpapier** **SIC-Team\_x.docx** gemäß Teilnahmebedingungen
2. Erstellen Sie eine **Präsentation** **A3-Team\_x.pptx** mit den wichtigsten Daten des Ideenpapiers

**Upload** der beiden Dateien auf WueCampus2 bis 09.05.2020 / 23:55

**Präsentation am 11.05.2020 durch alle aktuellen Projektleiter**  
**Dauer: 5 – 7 Minuten**

# Wie geht es weiter?

---

**15.05.2020, 10:00 Uhr**

- ▶ Gastvortrag von Frau Tanja Golly (SFT): Business Planning
- ▶ Elevator-Pitches max. 2 Minuten der aktuellen Projektleiter
- ▶ Besprechung der Ideen

**bis 19.05.2020, 23:55 Uhr**

- ▶ Ideenpapiere optimieren und hochladen auf WueCampus

**bis 22.05.2020**

- ▶ Einreichung der Ideenpapiere bis 22.05.2020 per E-Mail an Frau Golly [tanja.golly@uni-wuerzburg.de](mailto:tanja.golly@uni-wuerzburg.de)

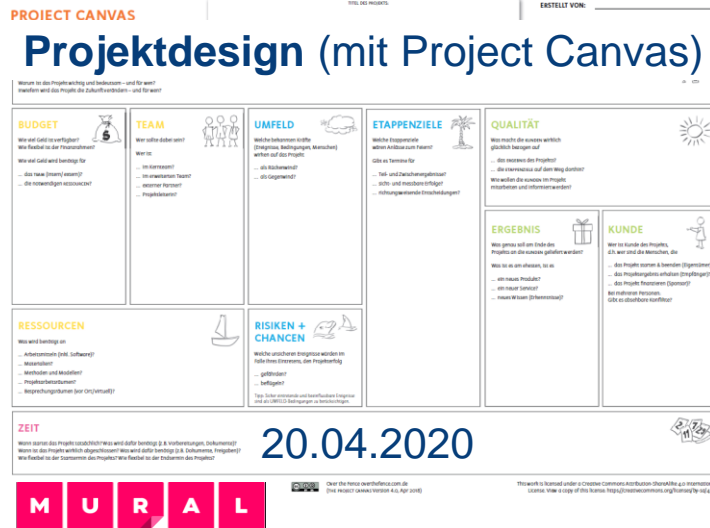
Ggf. Einladung zur Phase 2

# Zeit für eine Kaffeepause

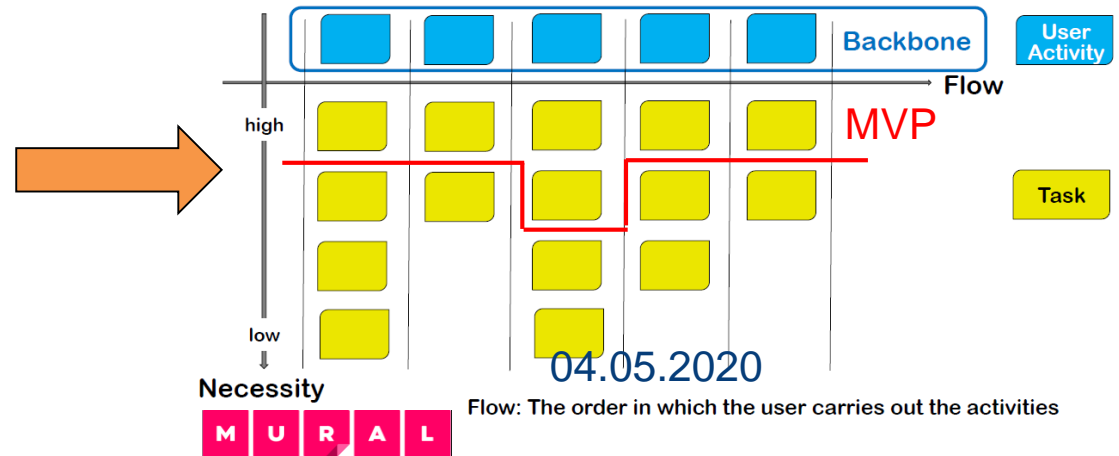




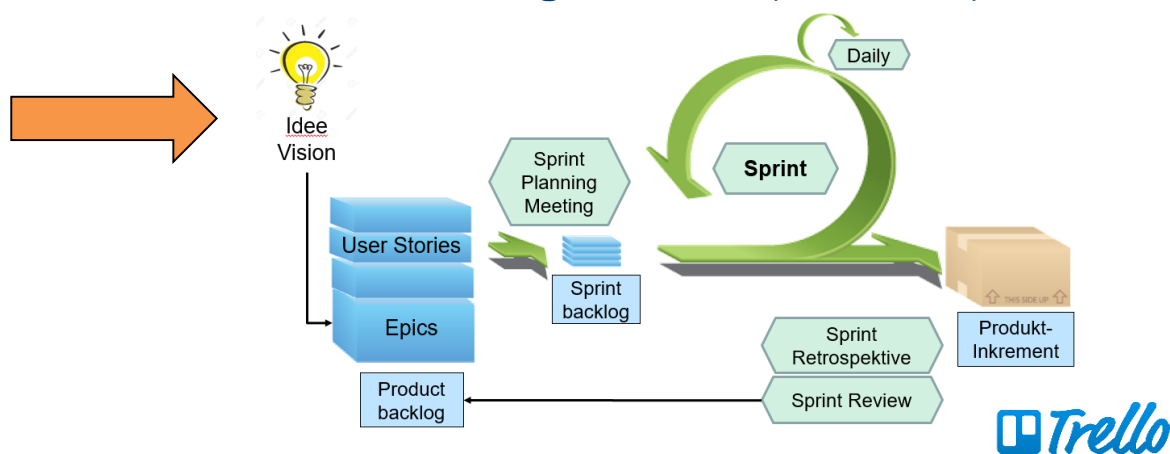
# Unser Weg zum MVP



## MVP Entwurf (über LSU / Story Map)



## Erstellung des MVP (mit Scrum)





Gedränge = Enge Zusammenarbeit im Team

# SCRUM: WIEDERHOLUNG UND ERGÄNZUNG

# Überblick: Scrum

## 3 Rollen

- Product Owner
- Scrum Master
- Entwicklungsteam

## 3 Artefakte

- Product Backlog
- Sprint Backlog
- Inkrement

## Transparenz der Artefakte

- Definition of Done

## 5 Ereignisse (Events, Meetings)

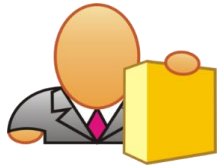
- Sprint
  - Sprint Planning Meeting
  - Daily Scrums Meeting
  - Sprint Review Meeting
  - Sprint Retrospektive Meeting

## 5 Werte

- Selbstverpflichtung
- Mut
- Fokus
- Offenheit
- Respekt

# 3 Rollen in Scrum

## Scrum Team



### **Product Owner (Einzelperson):**

verantwortlich dafür, dass die betriebswirtschaftlich richtigen Dinge gemacht werden



### **Scrum Master (Einzelperson):**

sorgt für einen reibungslosen Scrum Prozess



### **Entwicklungsteam:**

erledigt die Arbeiten in Selbstorganisation

### **Weitere Rollen im Umfeld:**

- Management des Auftragnehmers
- Kunde (Auftraggeber)
- Anwender/User

# Product Backlog



**Product Backlog:** Priorisierte Liste, die alle Funktionalitäten und Eigenschaften umfasst, die das Produkt enthalten soll

- Jeder Eintrag sollte wertvoll für die Anwender des Produktes oder den Kunden sein
- Eintragungen mit der höchsten Priorität werden als erste im Sprint umgesetzt
- Vor Beginn jedes Sprints findet eine neue Priorisierung statt
- **Product Backlog entwickelt sich mit dem Produkt und dessen Einsatz weiter**
- Verantwortlich: Product Owner

# Formulierung von User Stories

## INVEST – PRINZIP

### Independent

- Unabhängig voneinander

### Negotiable

- Verhandelbar

### Valuable

- Wertvoll für den Kunden

### Estimatable

- Aufwand und Kosten sind schätzbar

### Small

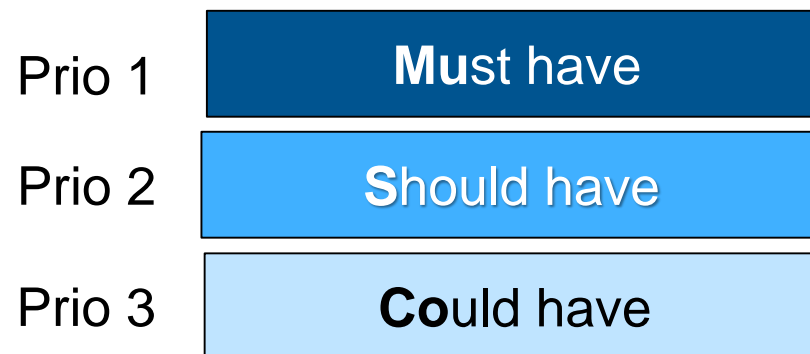
### Testable

- Testbar

# Priorisierung von User Stories

## MuSCow – Prinzip

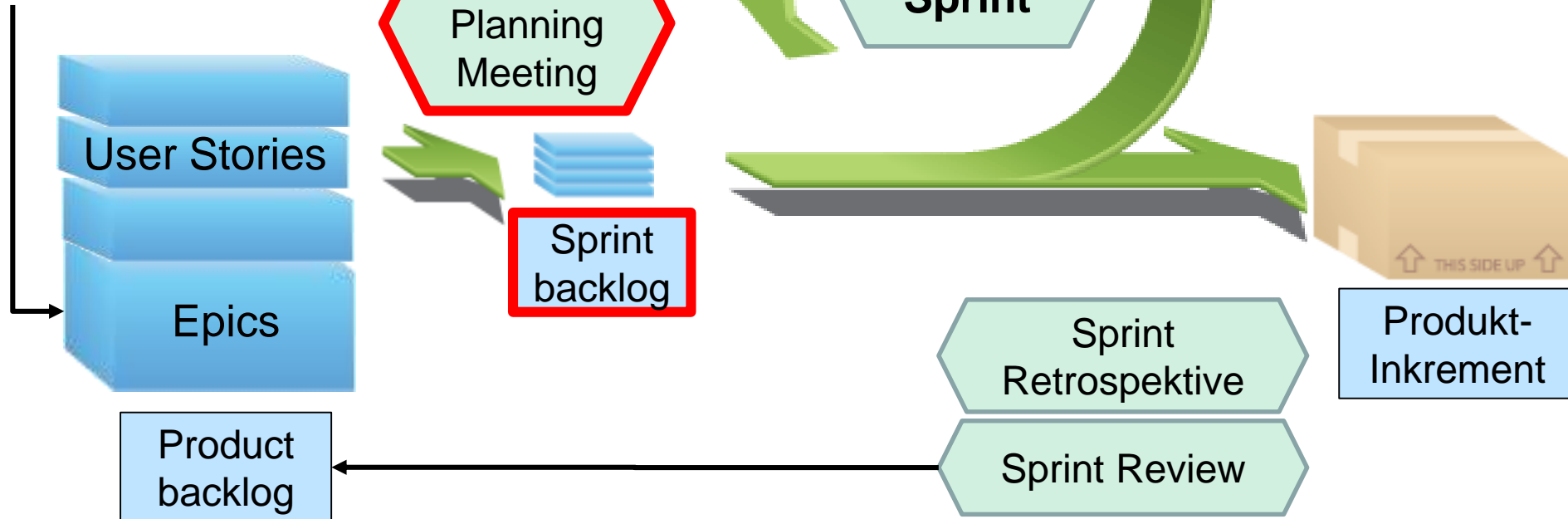
- **Must have** – zwingend notwendige User Story
- **Should have** – wichtige User Story, aber ohne sie ist der Projektgegenstand verwendbar
- **Could have** – User Story wird nur implementiert, falls noch genügend Zeit und Geld vorhanden ist
- **Won't have** – dient der Abgrenzung, indem spezifiziert wird, was nicht implementiert wird



# Scrum Prozess: Sprint Planung und Sprint Backlog



Idee  
Vision





# Schätzungen mit Story Points

- ▶ Scrum kennt keine Aufwandsschätzung von User Stories in Personentagen
- ▶ Die **Größe der User Stories** wird geschätzt. Diese ist ein Maß für die Komplexität einer User Story. Ihre Maßeinheit ist **Story Point**.
  - Hat eine User Story doppelt so viele Story Points wie eine andere User Story, so hat sie die doppelte Komplexität
- ▶ Bei der Schätzung der Größe einer User Story werden **alle Aufgaben** (Implementierung, Test, Dokumentation) **mit einbezogen**
- ▶ Vorgehen
  - Eine mittelgroße User Story wird als **Basisgröße** gewählt
  - Die **Größen der anderen Stories** werden **in Relation** dazu gesetzt
- ▶ Vorteile
  - Schnellere Schätzung, da unabhängig von konkreten Aufwänden
  - Komplexität und Größenverhältnisse der Features im Projekt werden sichtbar

# Schätzung der Backlog Items mit Planungspoker (1/2)

- Die Größe der User Stories (Backlog Items) werden vom Team geschätzt
- Jede User Story erhält **Story Points**, die die relative Größe (Funktionsumfang und Komplexität) im Vergleich zu den anderen widerspiegelt  
→ Story Points sind **relative Vergleichsgrößen** der einzelnen Backlog Items untereinander – keine absoluten Schätzgrößen, wie Personentage

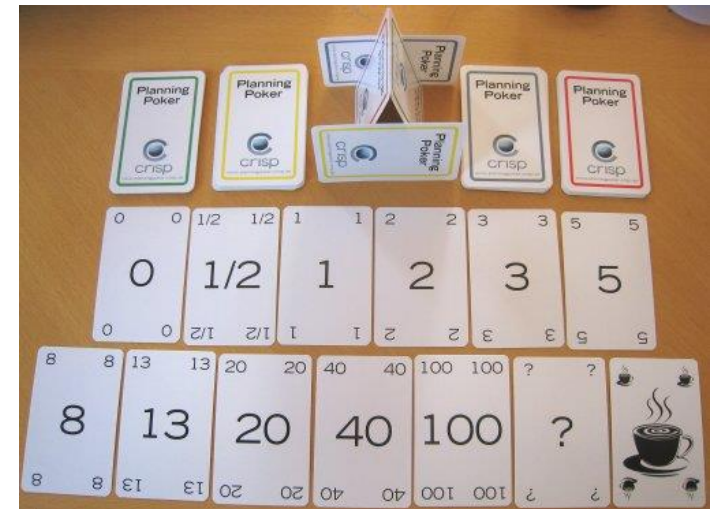
- **Planning Poker Karten**

Kartenwerte:

0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100,

? steht für unsicher und

**Kaffeetasse** steht für Pause erwünscht



# Schätzung der Backlog Items mit Planungspoker (2/2)

## Vorgehen

- Jedes Team Mitglied erhält einen Satz von 13 Poker Karten
  - Das Team identifiziert eine **mittelgroße Referenz-Story** und bewertet diese mit 5 Story Points
1. Product Owner erläutert eine Story
  2. Jedes Teammitglied schätzt die Größe für diese Story im Vergleich zur Referenzstory und legt eine entsprechende Karte verdeckt auf den Tisch (z.B. etwa 4 x so groß → „20“)
  3. Alle Karten werden aufgedeckt und verglichen
  4. Stimmen diese nicht überein, so begründen die Teilnehmer mit der höchsten und der niedrigsten Karte ihre Bewertungen
  5. Ggf. gibt es weitere Schätz- und Diskussionsrunden – bis Konsens vorliegt
  6. Weiter mit 1. – bis alle Backlog Items geschätzt sind

# Planning Poker – online

1. Jedes Teammitglied schätzt die Größe für die Story im Vergleich zur Referenzstory und hält die entsprechende Karte mit der Rückseite ins Bild.



2. Wenn alle fertig sind, drehen alle auf Kommando diese Karten um und zeigen ihre Werte.



3. Weiteres Vorgehen wie in der Präsenzversion

# Planning Poker – online-Tool



hatjitsu

<https://tools.wmflabs.org/hatjitsu/>

Distributed scrum planning poker for estimating agile projects.

First person to **create** the room is the moderator. **Share** the url or room number with other team members to **join** the room.

Enter room number

Join room

or

Create new room

© [UBXD](#) 2012 - 2013; Built by [@Jörg](#) & [@Rich](#). [Feedback?](#)



# Beispiel: Product Backlog mit priorisieren Items und Story Points

Product Backlog Items: <b>Must Have</b>		Story Points
US1	Bürger kann Anliegen bei Stadt einreichen	20
US2	Bürger kann BürgerBot an Logo identifizieren	13
US3	Mitarbeiter erhält kategorisierte Anliegen	8
US4	Bürger kann Stadt über Anliegen informieren	13

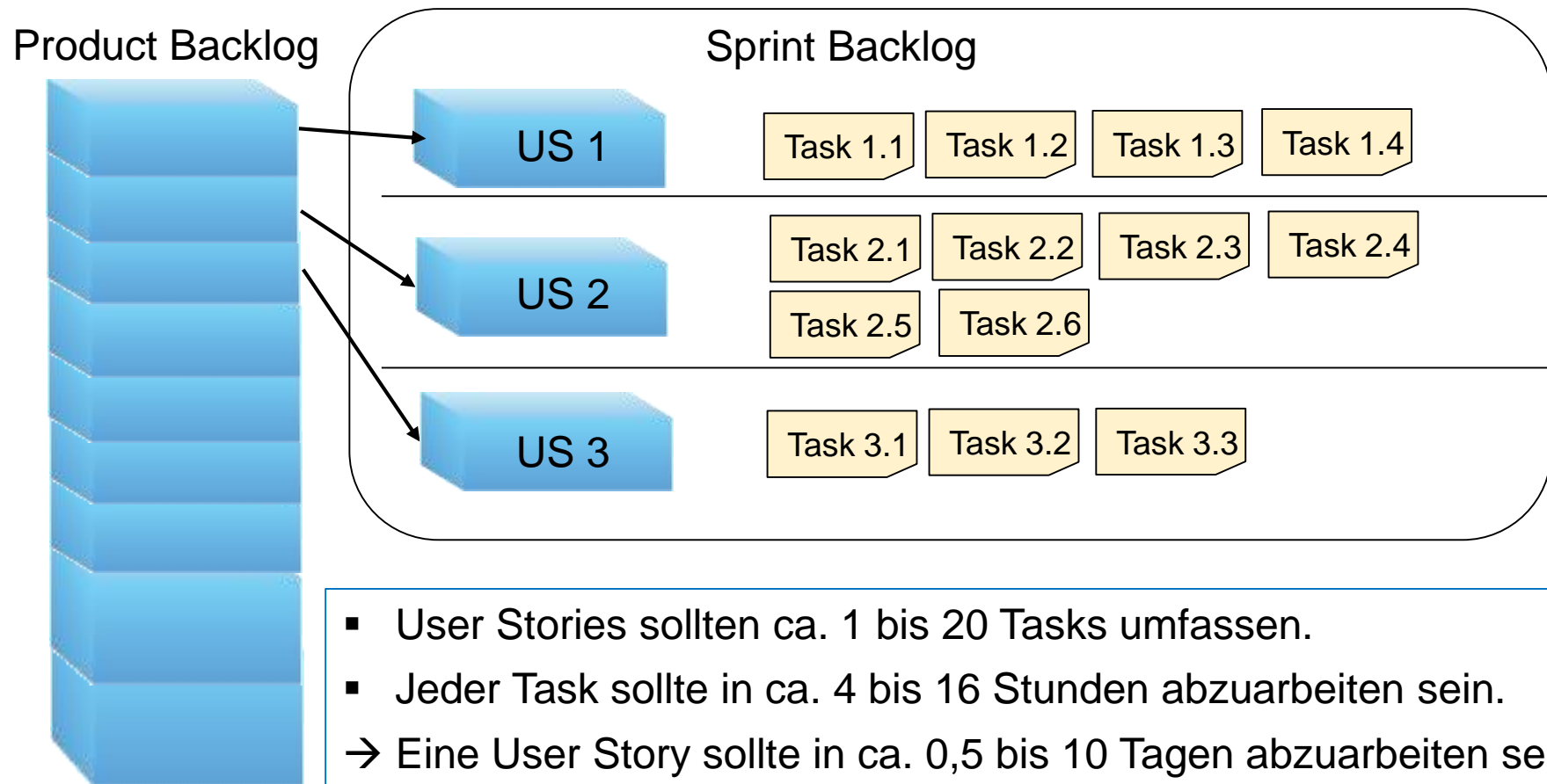
Product Backlog Items: <b>Should Have</b>		Story Points
US5	Mitarbeiter erhält übermittelte Anliegen	8
US8	Bürger wird per Poster über Kontaktmöglichkeit informiert	13
US6	Mitarbeiter erhält übermittelte Anliegen	8
US9	Bürger kann Anliegen per WhatsApp übermitteln	13
US7	Bürger kann Anliegen übermitteln	8
US10	Bürger kann Bilder zum Anliegen übermitteln	8

Product Backlog Items: <b>Could Have</b>		Story Points
US11	Mitarbeiter erhält übermittelte Anliegen	8
US12	Mitarbeiter erhält Namen des Bürgers	5
US13	Bürger kann Anliegen per Facebook übermitteln	13
US14	Bürger kann Status seines übermittelten Anliegens abfragen	13
US15	Bürger kann Anliegen teilen und " liken "	8
US16	Mitarbeiter kann Karte mit Anliegen anzeigen lassen	20
	Anreizsystem zur Bürgerbeteiligung nutzen	40

# Sprint Planning: Stories → Tasks

1. Auswahl der Einträge aus dem Product Backlog, um Sprint-Ziel zu erreichen

2. Ableiten der erforderlichen Tasks (Umsetzungsplan)



# Beispiel: User Stories in Tasks zerlegen

US1 Bürger kann Anliegen bei Stadt einreichen

20

Als Bürger  
Kommunikation  
Anliegen

US4 Bürger kann Standort zum Anliegen übermitteln

13

Als Mitarbeiter der Stadt Würzburg möchte ich die von den Bürgern  
Standort zum Anliegen  
präziser bestimmen

US5 Mitarbeiter kann Anliegen im Ticketsystem einsehen

3

Als Mitarbeiter der Stadt Würzburg möchte ich die von den Bürgern  
erfassten Anliegen in unserem vorhandenen Ticketsystem einsehen und  
bearbeiten können, damit meine Arbeit effizienter wird.

Task 1.1 D

Task 1.2 C

Task 1.3 R

Task 1.4 D

Task 1.5 M

Task 4.1 Ex

Task 4.2 M

Task 4.3 D

Task 4.4 Lo

Task 5.1 Einrichten der Mail-Schnittstelle



# Workshop: User Stories des MVP in Tasks zerlegen

---

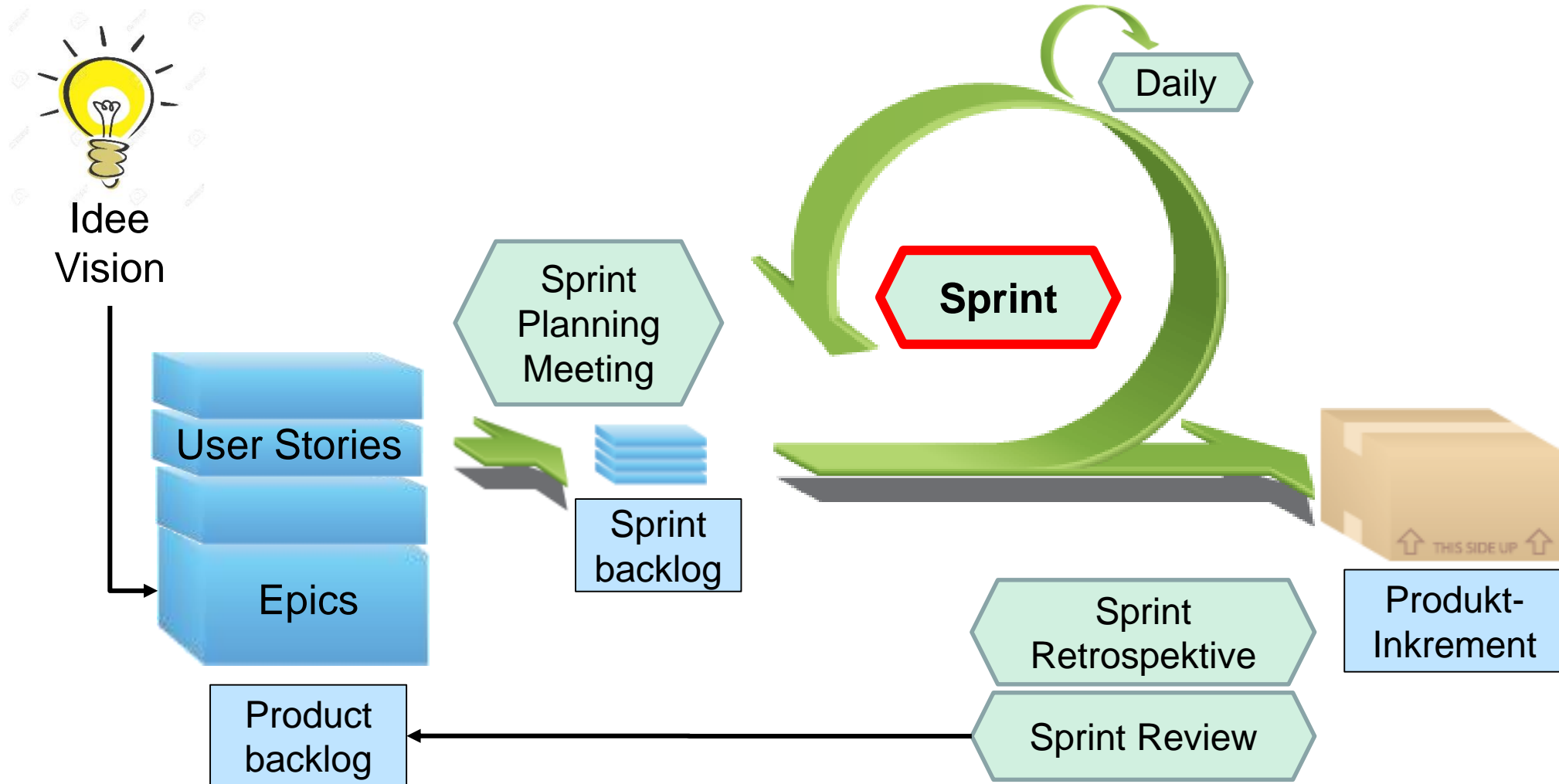
Wählen Sie **die beiden wichtigsten User Story Ihres MVP** aus und zerlegen Sie diese in alle **Tasks**, die erforderlich sind, damit die Story technisch umgesetzt ist

Start: Uhr

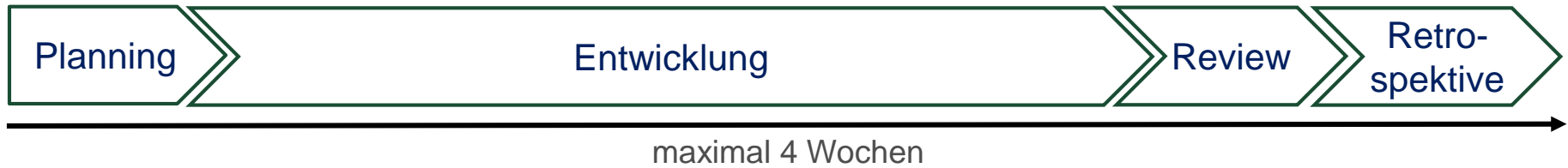
Ende: Uhr

(Dauer: 15 Minuten)

# Scrum Prozess: Sprint



# Sprint



- ▶ Alle Sprints innerhalb eines Entwicklungsvorhabens sollten die gleiche Dauer haben (max. 1 Monat; üblich auch 1-, 2-, 3- oder 4-Wochen)
- ▶ Der neue Sprint startet sofort nach Abschluss des vorherigen Sprints
- ▶ Sprint-Inhalte
  - Sprint Planning
  - Daily Scrums
  - Entwicklungsarbeit
  - Sprint Review
  - Sprint Retrospektive
- ▶ Jeder Sprint hat ein **Ziel**, was gebaut werden soll, einen **Entwurf** und einen **flexiblen Plan**, der die Umsetzung, die Arbeit und das resultierende Produktinkrement in die richtige Richtung lenkt

# Status der Erledigung der Tasks wird auf einem Task Board oder mit einem Tool (z.B. Trello) visualisiert

	To Do	In Process	To Verify	Done
<b>Sprint Backlog</b>	<b>Offene Tasks</b>	<b>In Arbeit</b>	<b>Im Test</b>	<b>Erledigt</b>
<b>User Story 1</b>	Task 1.3 Task 1.4 Task 1.6	Task 1.2 Task 1.5	Task 1.7	Task 1.1
<b>User Story 2</b>	Task 2.3 Task 2.4	Task 2.1		Task 2.2
<b>Fehlerbehebung</b>		Task F.3	Task F.2	Task F.1

Mitarbeiter des Entwicklungsteams greifen sich selbständig Tasks und starten mit der Bearbeitung



Jeder hat stets den aktuellen Überblick über den Fortschritt im Sprint

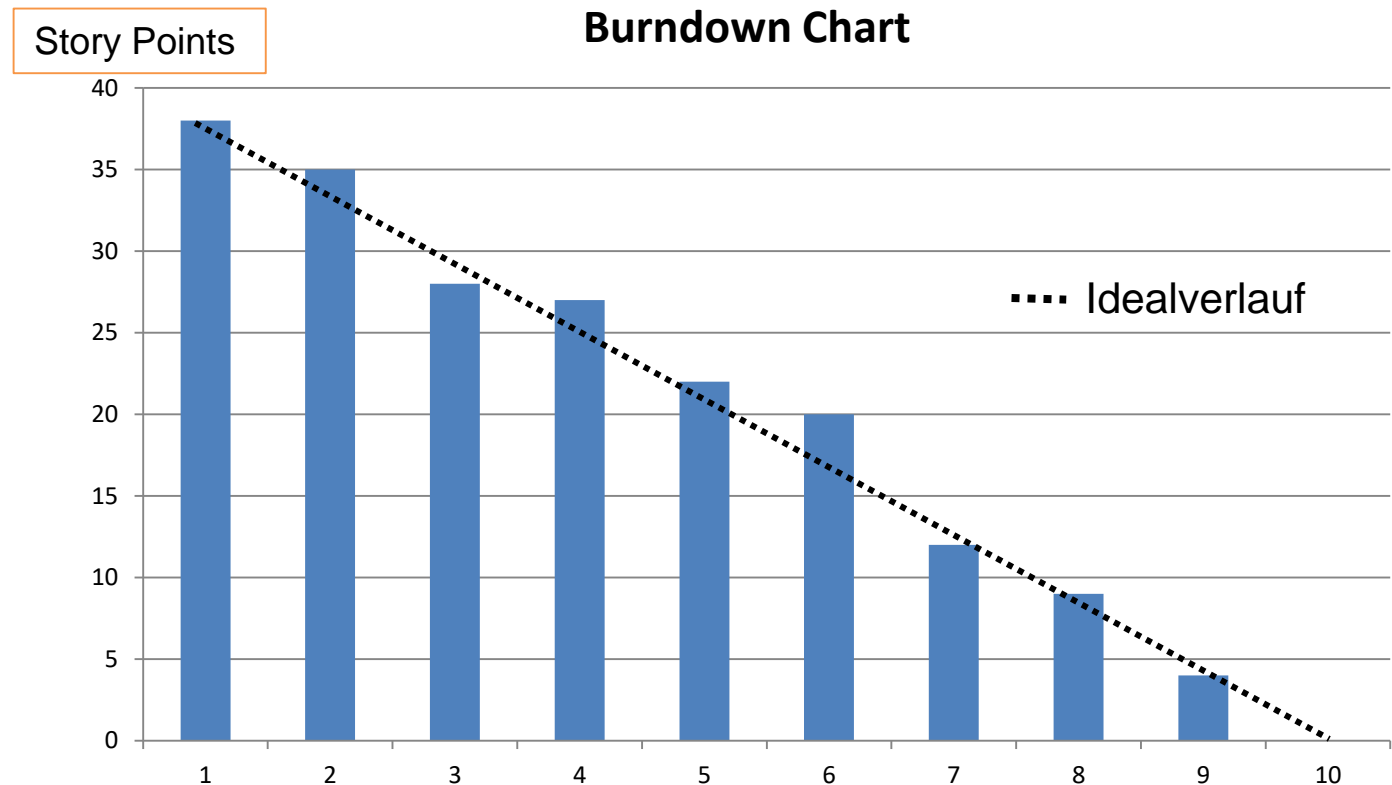
# Restliche Arbeit wird im Burndown Chart visualisiert

## Aufbau

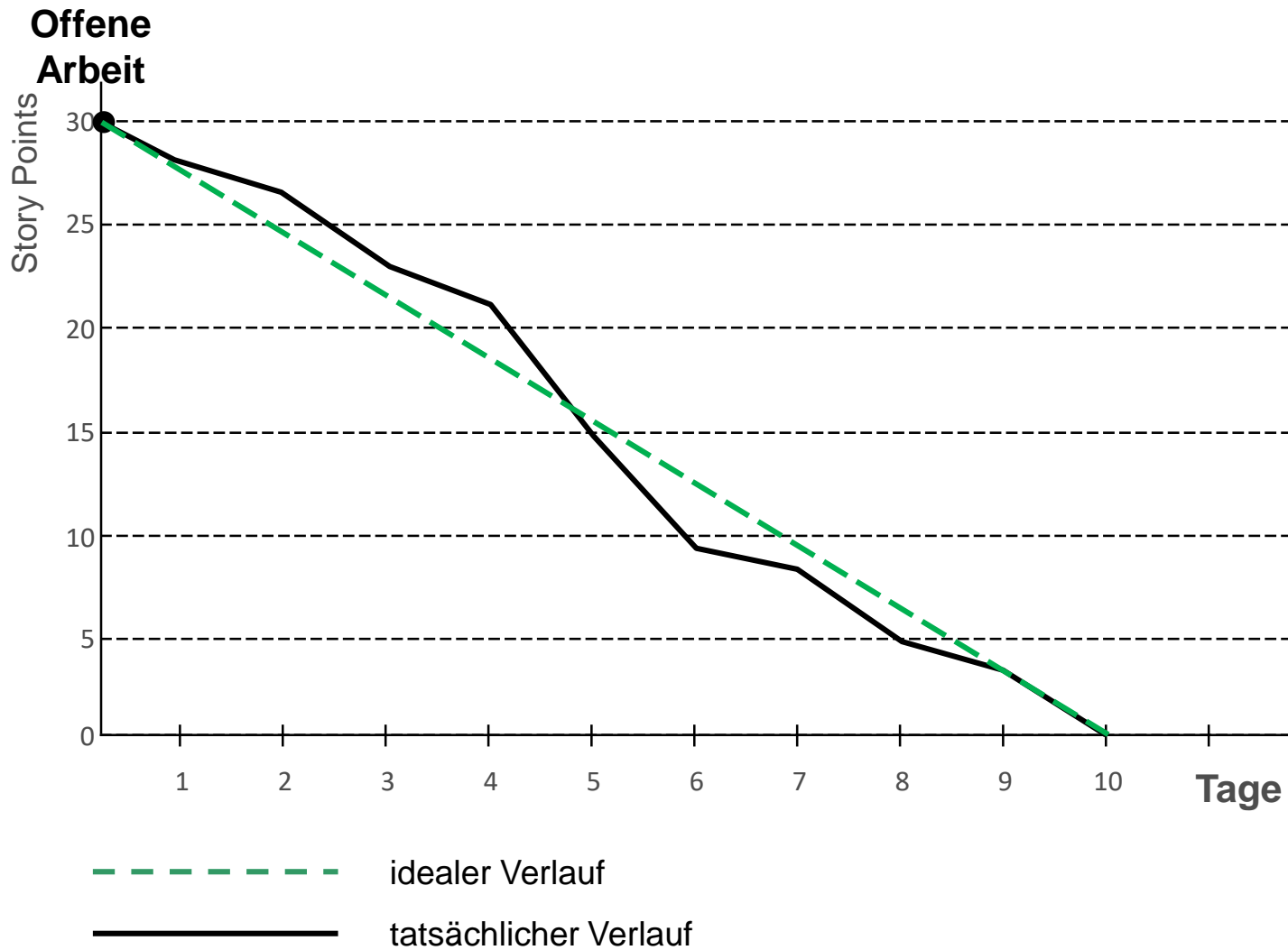
- x-Achse beschreibt Zeitverlauf (in Arbeitstagen)
- y-Achse die geschätzten Story Points der (noch) nicht erledigten Tasks

## Motivation

- Fortschritt wird transparent
- Erreichung des Sprint-Ziels abschätzbar

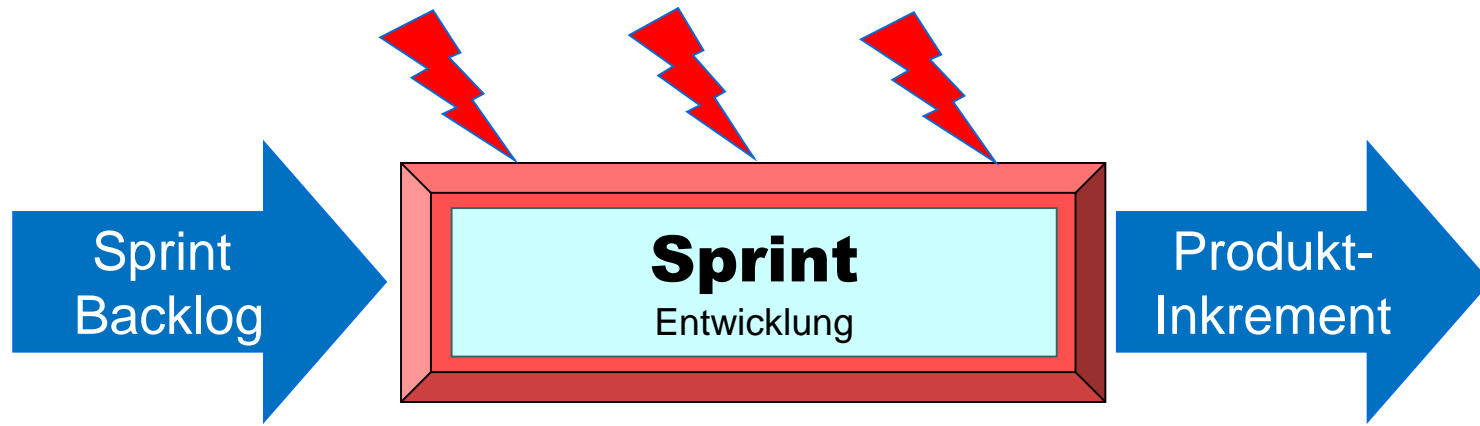


# Burndown Chart – als Kurve



# Grundregel: Keine Änderungen während eines Sprints!

## Änderungswünsche sind Tabu



- Änderungs- und Zusatzwünsche werden während des Sprints vom Product Owner gesammelt und dem Product Backlog zugeführt
- Sie können erst beim nächsten Sprint – falls entsprechend priorisiert – Berücksichtigung finden
- Während eines Sprints kann der **Anforderungsumfang** auf Basis neuer Erkenntnisse zwischen Product Owner und Entwicklungsteam neu ausgehandelt werden.  
Dabei darf der **Qualitätsanspruch** nicht geschmälert werden.

# Sprint Abbruch (eher unüblich)

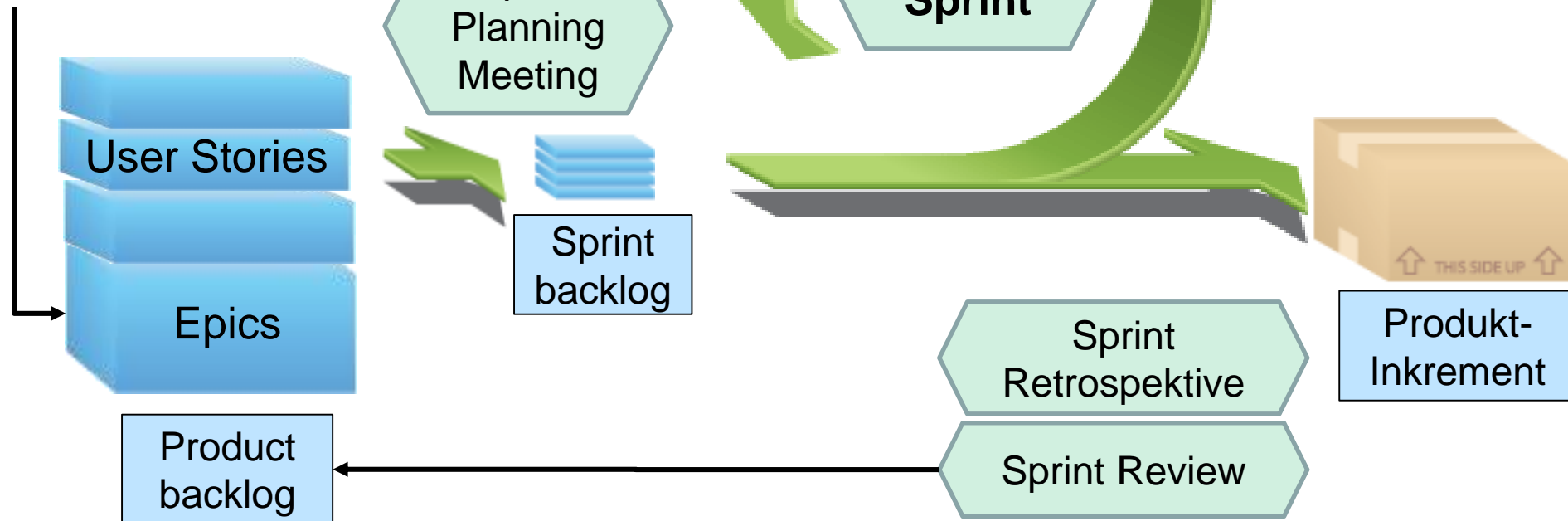
- ▶ **Nur der Product Owner hat die Berechtigung für einen vorzeitigen Abbruch eines Sprints**
- ▶ **Mögliche Abbruch-Gründe**
  - Sprint-Ziel wird obsolet, z.B. durch Änderungen im Wettbewerb
  - In der Regel sollte ein Sprint dann abgebrochen werden, wenn die Fortführung unter den gegenwärtigen Umständen keinen Sinn mehr macht.
- ▶ **Aktivitäten bei Sprint Abbruch**
  - Alle bereits abgeschlossenen und "Done" Product-Backlog-Einträge werden begutachtet
  - Wenn ein Teil der Arbeit potenziell auslieferbar ist, wird sie vom Product Owner meistens abgenommen
  - Alle unvollständigen Product-Backlog-Einträge werden neu geschätzt und wieder in das Product Backlog aufgenommen



# Scrum Prozess: Daily Scrum



Idee  
Vision



# Scrum Ereignis: Daily Scrum

- Ziel: Austausch über Projektfortschritt und Hindernisse
- Regeln:
  - täglich (Time Box: 15 Minuten)
  - stets zur gleichen Zeit und am gleichen Ort
  - im Stehen
  - Scrum Master moderiert
- Jeder beantwortet folgende drei Fragen:
  - *Was habe ich zur Erreichung des Sprint-Ziels seit dem letzten Daily gemacht?*
  - *Welche Arbeiten werde ich heute erledigen, um dem Entwicklungsteam bei der Erreichung des Sprint-Ziels zu helfen ?*
  - *Welche Hindernisse stehen mir oder dem Entwicklungsteam im Weg?*

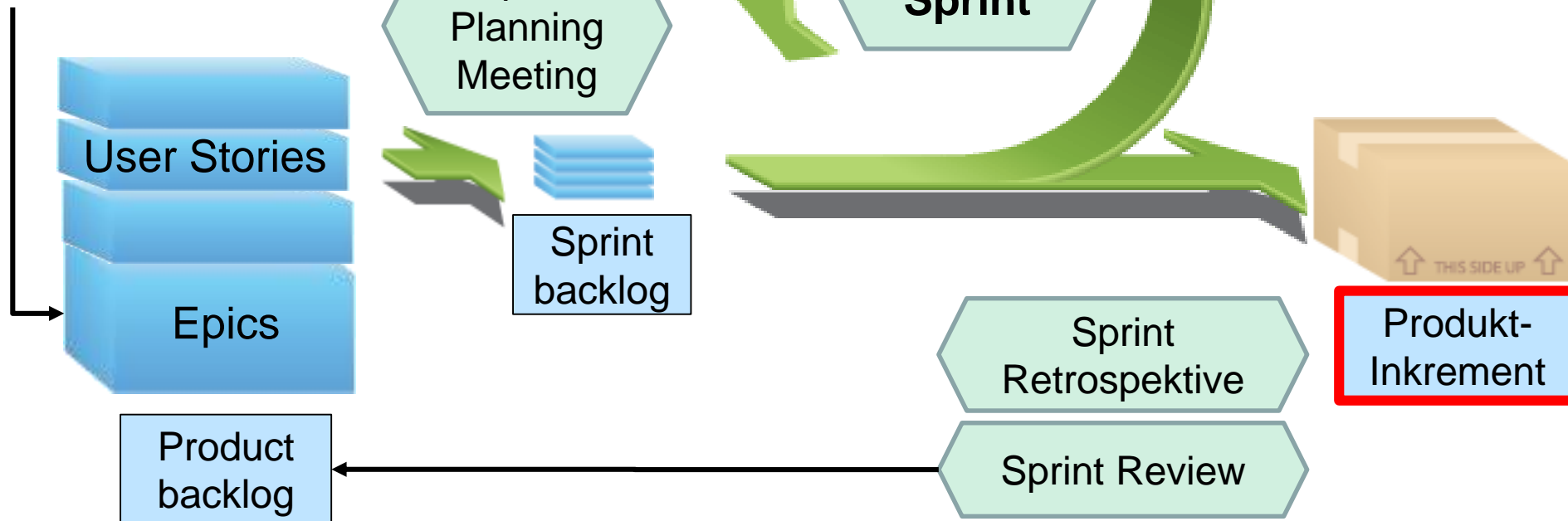
- Teilnehmer:



# Der Scrum Prozess: Produkt Inkrement



Idee  
Vision



# Produkt Inkrement

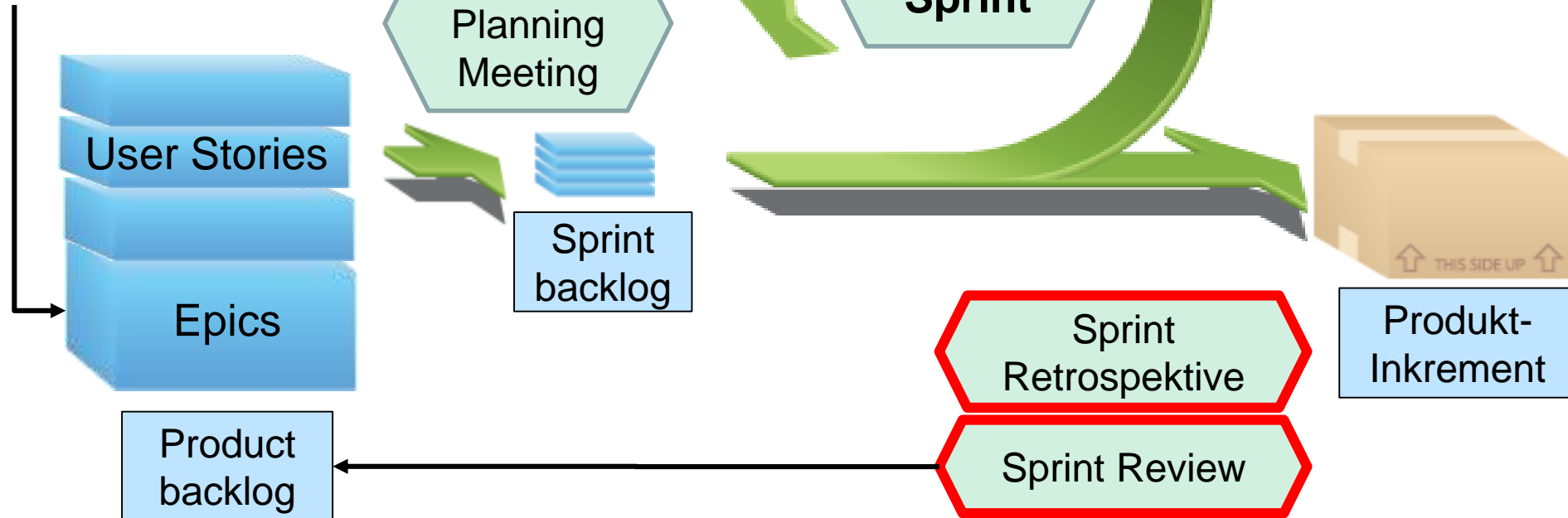
- ▶ Das Inkrement ist das Ergebnis aus allen in einem Sprint fertiggestellten Product-Backlog-Einträgen und dem Resultat der Inkremente aller früheren Sprints.
- ▶ Am Ende eines Sprints muss das neue Inkrement fertig ["Done"] sein; das heißt es muss in einem verwendbaren Zustand sein und die Definition of Done des Teams erfüllen.
- ▶ Das Inkrement ist ein Schritt in Richtung einer Vision oder eines Ziels.
- ▶ Es muss auch dann im einsatzfähigen Zustand sein, wenn der Product Owner es aktuell noch gar nicht ausliefern will.



# Der Scrum Prozess: Sprint Review und Retrospektive



Idee  
Vision



# Sprint-Review Meeting

## Sprint-Review Meeting: **Ergebnisvorstellung und -abnahme**

- Team präsentiert die neuen Funktionalitäten des Sprints
- Time Box: max. 4 Stunden bei 4-Wochen-Sprint
- Keine Folien!
- Produkt Owner / Anwender: begutachten und entscheiden, welche der Funktionalitäten abgenommen werden
- Unvollständige Features werden nicht abgenommen, kehren zurück ins Product Backlog und werden neu priorisiert

▪ Teilnehmer:



präsentiert



moderiert





nimmt ab



Weitere  
Stake-  
holder

# Sprint-Review Meeting: Ablauf





## Meeting-Vorbereitungen

-  Scrum Master notiert, welche Features vollständig realisiert wurden
-  Team bereitet dazu eine Demonstration vor. Nur vollständig realisierte Features dürfen gezeigt werden

## Meeting-Verlauf

-  Team demonstriert die vollständigen und getesteten Features
-  Product Owner akzeptiert oder verwirft die Ergebnisse
- Nicht abgenommene Tasks / User Stories kehren ins Product Backlog zurück

## Meeting-Nachbereitungen

-   Scrum Master und Team ermitteln die „Velocity“ des Sprints
-   Product Owner und Team aktualisieren den Product Backlog und bereiten nächstes Sprint-Planning-Meeting vor

# Velocity (Entwicklungsgeschwindigkeit)

- ▶ **Velocity = Anzahl der Story Points der fertigen und abgenommenen Stories** des vorangegangenen Sprints
- ▶ Velocity: dynamischer Wert
- ▶ Typische Velocity Entwicklung:

Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6
85 SP	75 SP	90 SP	120 SP	110 SP	120 SP

- ▶ Durchschnitt: 100 SP (Sprint 1-6); Durchschnitt: 110 (Sprint 3-6)
- ▶ **Nutzen** der Velocity
  - Nächster Sprint: Einschätzung einer angemessene Anzahl an Story Points
  - Gesamt-Entwicklungsdauer des Produktes: Hochrechnung
  - Beispiel zur Hochrechnung
    - Sprintdauer: 2 Wochen; Velocity 60; Product Backlog: 195 SP
    - Geschätzte Entwicklungsdauer: 8 Wochen (mit 4 Sprints)

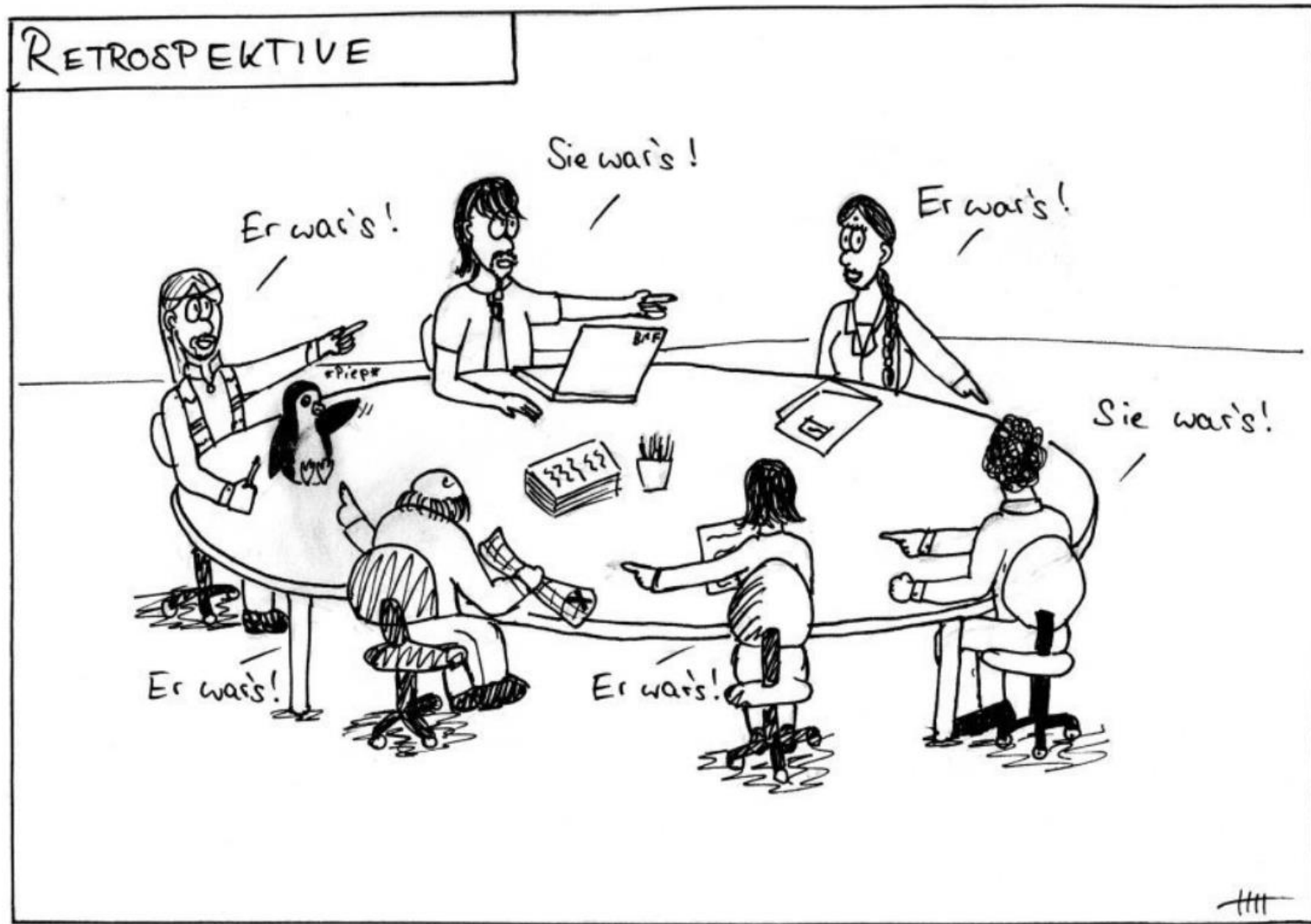


# Sprint Retrospektive (Retro): Interne Nachbetrachtung

- ▶ **Ziel: Verbesserung des Arbeitsprozesses** für Folge-Sprints
- ▶ Findet jeweils am Ende eines Sprints und vor dem nächsten Sprint Planning statt – Time Box von max. 3 Stunden bei 1-monatigem Sprint
- ▶ Inhalte: Konstruktiv-kritische Nachbetrachtung, wie der vergangene Sprint in Bezug auf die beteiligten Personen, Zusammenarbeit Beziehungen, Prozesse und Werkzeuge verlief
  - Was sollten wir neu probieren? (Start)
  - Was sollte verbessert werden? (stopp)
  - Was hat gut funktioniert? (Continue)
- ▶ SM erstellt eine Liste mit den priorisierten Verbesserungsmaßnahmen (**Impediment Backlog**) und kümmert sich um deren Umsetzung
- ▶ Teilnehmer:



# Sprint Retrospektive: So nicht!



# Formate für Retrospektiven

- ▶ Wie führt man spannende Retrospektiven durch?
  - Verschiedene Formate verwenden!
- ▶ Retromat: weit über 100 verschiedene Formate für Retrospektiven
  - <https://retromat.org/en/?id=84-126-41-99-83>
    - Start – Stopp – Continue
    - I like – I wish



Photo by Simon Sablowski (atness, ...) Let the group choose 4.

# Definition of „Done“ (DoD)

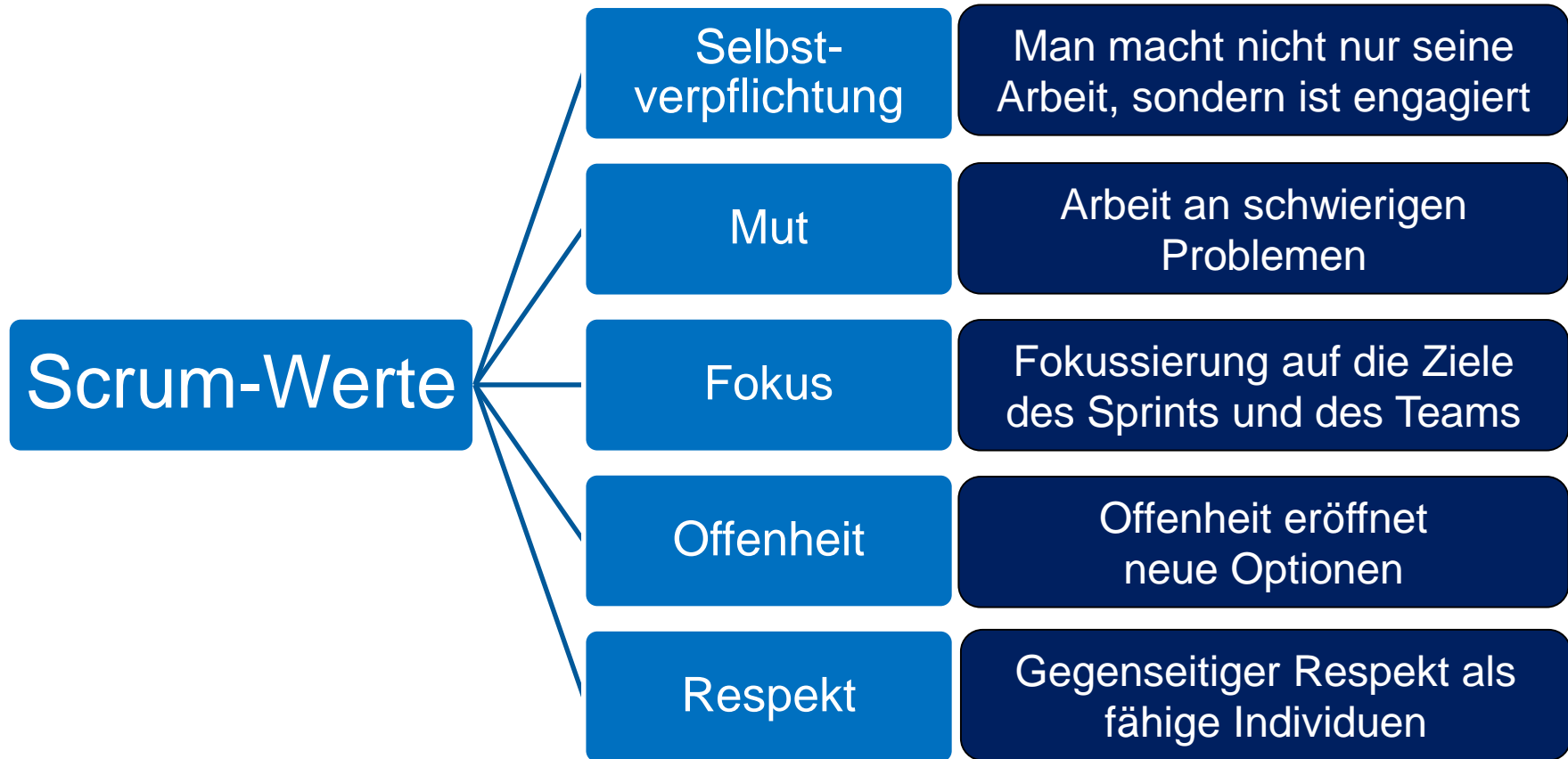
- ▶ DoD beschreibt die **Abnahmekriterien** für die Backlog Items
- ▶ DoD schafft **Transparenz**
  - Alle Teammitglieder und Stakeholder entwickeln ein gemeinsames Verständnis, wann eine Arbeit erledigt („Done“) ist
- ▶ Aktuelles DoD gilt für den aktuellen Sprint und wird bei der Retrospektive überprüft und ggf. weiterentwickelt
- ▶ Jedes Inkrement ist additiv zu allen früheren Inkrementen und gründlich getestet, um sicher zu stellen, dass alle Inkremente gemeinsam funktionieren

# Time-Boxing und Scrum Ereignisse (Events)

- **Alle Ereignisse haben einen festen Zeitrahmen [Time Box],**
- Vorteil: Meetings konzentrieren sich aufs Wesentliche, beginnen und enden pünktlich
- Scrum Master kümmert sich um die Einhaltung der Zeitlimits

Ereignis	Time-Box	Anmerkungen
<b>Sprint</b>	<b>1 Monat</b>	Alle Sprints innerhalb eines Projektes sollten die gleiche Dauer haben
<b>Sprint Planning</b>	<b>8 h</b> bei 1-monatigem Sprint	Für kürzere Sprints wird in der Regel ein kürzerer Zeitrahmen veranschlagt
<b>Daily Scrum</b>	15 Minuten	Täglich
<b>Sprint Review</b>	<b>4 h</b> bei 1-monatigem Sprint	Für kürzere Sprints wird in der Regel ein kürzerer Zeitrahmen veranschlagt
<b>Sprint Retrospective</b>	<b>3 h</b> bei 1-monatigem Sprint	

# 5 Scrum-Werte



---

# AUFGABE 4

# Aufgabe 4: Product Backlog und Sprint Planning

1. Formulieren Sie alle User-Anforderungen des MVP als **User-Stories**
2. Erstellen Sie ein **Product Backlog** Ihres Projektes
3. **Priorisieren** Sie die Backlog Items nach dem MuSCo Prinzip
4. **Schätzen** Sie die Größe (Komplexität) aller Backlog Items in Story Points
5. Leiten Sie aus allen **MVP-User Stories** die zu deren Erledigung **notwendigen Tasks** ab
6. **Reflektieren** Sie Ergebnisse und Prozesse

Erstellen Sie eine Präsentation mit den Ergebnissen von 1. - 6.

**Upload** auf WueCampus2 bis 16.05.2020 / 23:55: [A4-Team\\_x-Scrum.pptx](#)

**Präsentation** am 18.05.2020 durch die aktuellen PL der Teams 2 und 5  
Dauer: 8 – 10 Minuten



# Feedback

## Was mochte ich an der Vorlesung?

Wiederholung scrum

Präsentationen der Teams

Wiederholung und Vertiefung

Wiederholung des Stoffs sehr ausführlich

Fallbeispiele

Team-Präsentationen haben reibungslos funktioniert

Klare Zeitangaben, gute Struktur, (Scrum-)Wiederholung,

Wiederholung von scrum und ausblick, welche teilbereiche wir in den nächsten wochen noch weiter ausführen

Vergleich zu den nicht-agilen Verfahren

Kombination: Wiederholung und Neues (scrum) Tipps  
Ideenpapier Kombination:  
Wiederholung und Neues (scrum) Tipps Ideenpapier

Präsentationen waren im zeitl. Rahmen und alle gut

Motivierte Teamleads

# Feedback

## Was wünsche ich mir?

Mehr Gruppenarbeiten

Mehr Wiederholungssequenzen

Pausenlänge = 30 Minuten

Informationen zur bestmöglichen Klausurvorbereitung

Angaben zur Klausur. Ansonsten weiter so!

Scrum anwenden, weil auf theoretischer Ebene ist doch vieles noch unklar

Weiterhin Wiederholungen

Nächstes mal wirklich den zeitlichen Rahmen einhalten

Einhaltung der angegebenen Vorlesungszeit

Infos zum Zertifikat → bis wann Anmeldung etc.?

Ausführliches teambezogenes Feedback zum Ideenpapier

Weniger verschiedene, sich ähnelnde Beispiele; Breakout-Nutzung; Besprechung bzgl. Klausurinhalt;