

LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

FAKULTÄT FÜR MATHEMATIK, INFORMATIK UND STATISTIK
INSTITUT FÜR INFORMATIK

LEHRSTUHL FÜR DATENBANKSYSTEME
UND DATA MINING

Entwicklungsprozesse

SEP 2018

Tobias Lingelmann

2018-05-15

Wissenschaftliche Betreuer:

Daniel Kaltenthaler, Johannes Locher

Verantwortlicher Professor:

Prof. Dr. Peer Kröger



Inhalt

- Allgemeines
- Wasserfallmodell
- Agile Softwareentwicklung
- SCRUM
- Kanban
- eXtreme Programming



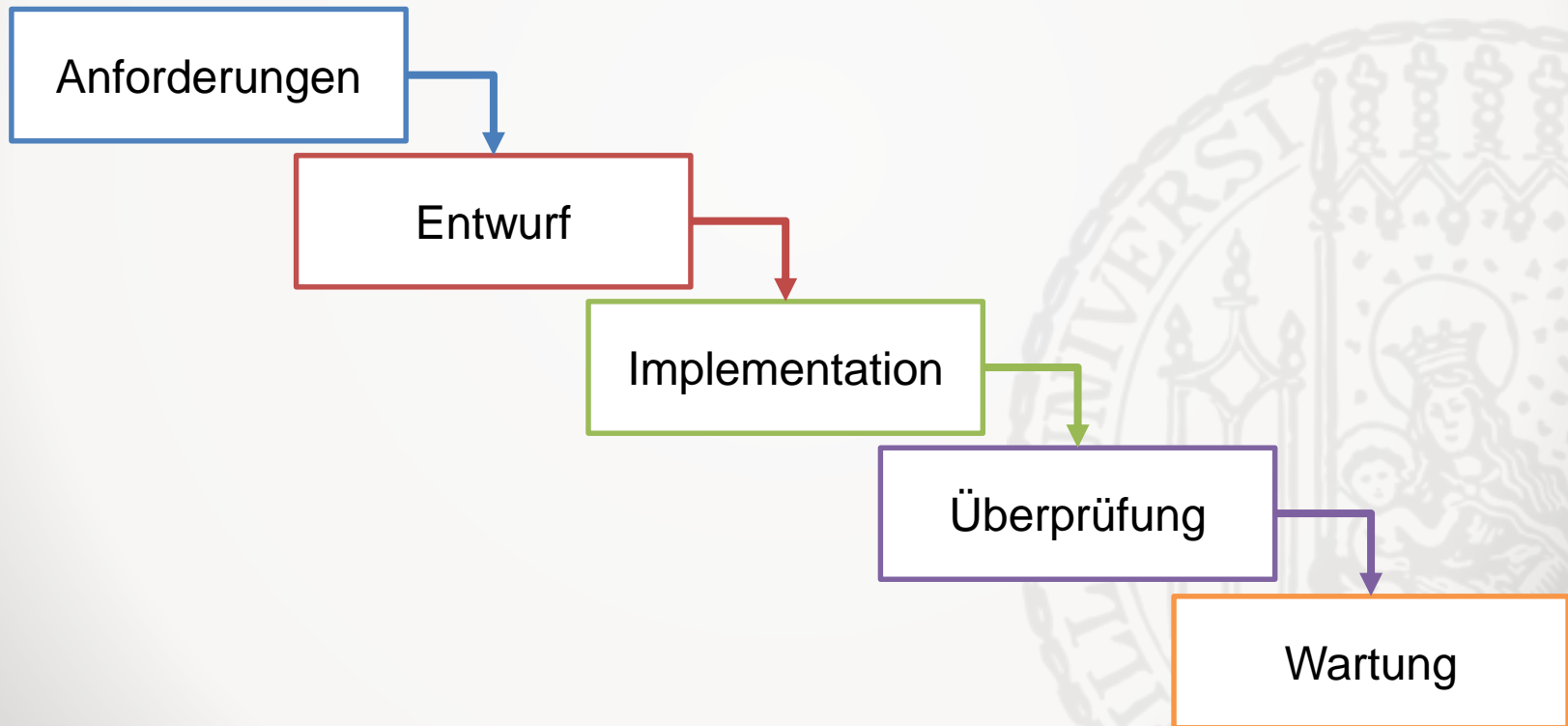
Allgemeines

- Ablaufmodell zum Entwickeln von Software
- Unterschiedliche Modelle: Hierarchisch, Inkrementell, Spiralförmig, Agil
- Verschiedene Methoden/Frameworks: Scrum, Kanban, Rapid-application develop.
- Viele Praktiken: Pair Programming, Stand-Ups, Continuous Delivery

Aufzählungen nicht vollständig!

Wasserfallmodell

- Lineares Vorgehensmodell
- Abfolge streng hierarchisch



Wasserfallmodell

Vorteile:

- Einfach zu verstehen mit natürlich identifizierbaren Meilensteinen.
- Gut, wenn Anforderungen und Aufgabenbereich klar definiert sind.
- Erlaubt klare Abschätzung von Kosten und Umfang.

Wasserfallmodell

Nachteile:

- Unflexibel gegenüber Änderungen
- Erst ganz am Ende steht ein fertiges Produkt zur Verfügung. (Spätes RoI)
- Fehler werden eventuell erst (zu) spät erkannt → Hohe Kosten!
- Anforderungen eventuell noch nicht bekannt oder werden falsch eingeschätzt.

Agile Softwareentwicklung

- Inkrementelles Vorgehen
- Einbindung des Kunden in den Entwicklungsprozess
- Ständiges Anpassen an eventuell veränderte Kundenwünsche
- Häufige Auslieferung von funktionierender (Teil)Software

→ Agiles Manifesto

Agile Softwareentwicklung

Vorteile:

- Flexibel gegenüber Änderungen
- Zeitnahes (=kostengünstiges) Reagieren auf geänderte Anforderungen möglich.
- Ständige Bereitstellung eines Produkt
- Besonders gut, wenn der Kunde selber noch nicht wirklich weiß, was er eigentlich benötigt.

Agile Softwareentwicklung

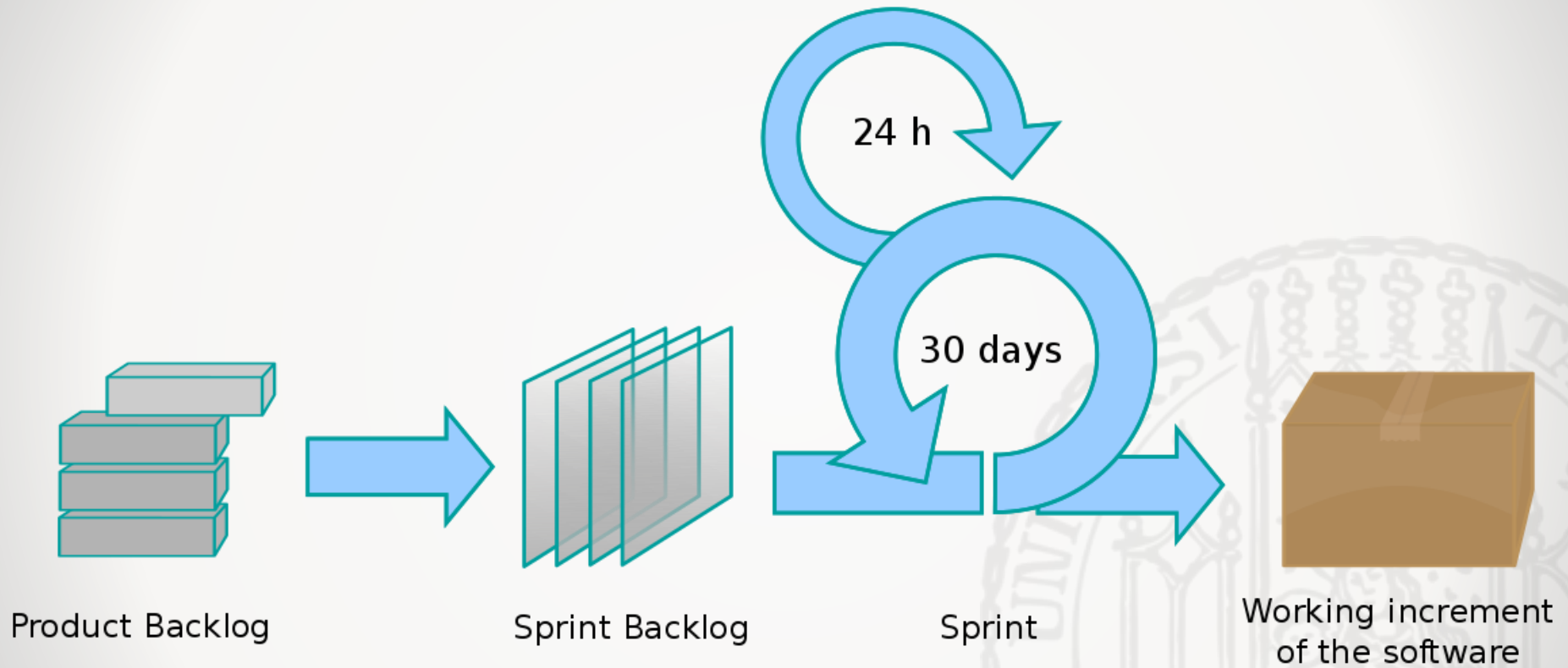
Nachteile:

- Kosten- und Zeitplanung schwierig(er)
- Wofür wird wie viel gezahlt?
- Mangelhafte Zusammenarbeit mit dem Kunden führt zu größeren Problemen.
- Fehlende Akzeptanz für agiles Vorgehen bei Entwicklern und Management kann zu unnötigem Verwaltungsaufwand führen.

Scrum

- Umsetzung von Features/User Stories
- Arbeit erfolgt in festen Zeitrahmen
- Kleine, selbstorganisierte Teams
- Teams sind interdisziplinär besetzt
- Klar definierte, regelmäßig stattfindende Aktivitäten
- Kontinuierliche Anpassung der Anforderungen und des Vorgehens

Scrum



[Quelle](#)

Scrum

- Rollen:
 - Product Owner
 - Entwicklerteam
 - Scrum master
- Artefakte:
 - Product Backlog
 - Sprint Backlog
 - Product Increment
- Aktivitäten:
 - Sprint Planning
 - Sprint
 - Daily Scrum
 - Sprint Review
 - Sprint Retrospektive

Kanban

- Visualisierung des Workflows durch „Kanban-Board“ mit Karteikarten/Haftnotizen
- Begrenzung der Tickets pro Spalte/Station
- Engstellen einfach ersichtlich
- Tickets müssen nicht zeitlich begrenzt sein.
- Teams dürfen spezialisiert sein

Kanban

Pool of Ideas	Feature Preparation		Feature Selected	User Story Identified	User Story Preparation		User Story Development		Feature Acceptance		Deployment	Delivered
Epic 431	3 - 10		2 - 5	30	15		15		8		5	Epic 294
Epic 478	In Progress	Ready	Epic 602		In Progress	Ready	In Progress	Ready (Done)	In Progress	Ready	Epic 694	Epic 386
Epic 562	Epic 444	Epic 662				Story 602-02	Story 602-06	Story 602-03	Epic 401	Epic 609	Epic 276	Epic 419
Epic 439	Epic 589		Epic 302	Story 302-03	Story 302-01	Story 302-07	Story 302-09	Story 602-04	Story 602-01	Epic 468	Epic 577	Epic 388
Epic 329	Epic 651		Epic 335	Story 302-02	Story 302-06	Story 302-08		Story 303-05	Story 302-04	Epic 362		Epic 287
Epic 287			Epic 512	Story 335-09	Story 335-10	Story 335-04	Story 335-05	Story 335-06				Epic 274
Epic 606	Discarded			Story 335-08	Story 335-01	Story 335-03	Story 335-02	Story 335-07				
	Epic 511	Epic 213		Story 512-04	Story 512-07	Story 512-02	Story 512-01					
	Epic 221			Story 512-05	Story 512-06	Story 512-03						

Policy
Business case showing value, cost of delay, size estimate and design outline.

Policy
Selection at Replenishment meeting chaired by Product Director.

Policy
Small, well-understood, testable, agreed with PD & Team

Policy
As per "Definition of Done" (see...)

Policy
Risk assessed per Continuous Deployment policy (see...)

eXtreme programming

- Teamarbeit an erster Stelle. (Code gehört allen, statt jedem „sein Teil“.)
- Kunde wird aktiv zur Mitarbeit aufgefordert
- Dokumentiert wird, was als sinnvoll erachtet wird.
- Viel Testen und Kontrollieren (automatische Tests, zeitnahes Code-Review durch z.Bsp. Pair Programming)
- Weniger „planen“, mehr „machen“

Linksammlung

- https://en.wikipedia.org/wiki/Software_development
- <https://agileinsights.wordpress.com/tag/agile-development/>
- <http://agilemanifesto.org/iso/de/manifesto.html>
- <https://toggl.com/developer-methods-infographic/>
- <https://realtimeboard.com/blog/choose-between-agile-lean-scrum-kanban/>
- <https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>
- <http://scrumguides.org/scrum-guide.html>
- <https://www.pmi.org/learning/library/agile-problems-challenges-failures-5869>