

Seminar: Softwareentwicklung in der Wissenschaft

Agile Softwareentwicklung

Benjamin Pöpel

Fakultät für Mathematik, Informatik und Naturwissenschaften

Fachbereich Informatik

Betreuer: Christian Hovy

23. Juni 2014

Inhaltsverzeichnis

- 1 Agile Softwareentwicklung
 - Konzept der agilen Softwareentwicklung
 - Agile Softwareentwicklung und Projektmanagement
 - Abgrenzung zu anderen Konzepten der Softwareentwicklung
- 2 Extreme Programming (XP)
 - Weitere Verfahren
- 3 Chancen & Risiken
- 4 Zusammenfassung
- 5 Literaturverzeichnis

Agile Manifest

Das agile Manifest enthält die folgenden Wertaussagen:

- *Menschen und Zusammenarbeit*
sind wichtiger als Prozesse und Werkzeuge.
- *Lauffähige Software*
ist wichtiger als umfangreiche Dokumentation.
- *Zusammenarbeiten mit Auftraggebern*
ist wichtiger als Vertragsverhandlungen.
- *Reagieren auf Änderungen*
ist wichtiger als das sture Befolgen eines Plans.

Agile Manifest

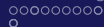
- Betonung der Interaktionen unter den beteiligten Personen des Projekts
- Schwierigkeit der Umsetzung der Wertaussagen
- Fokussierung auf Zielsetzungen und auf Behandlungen von sozialen und technischen Problemen



Agile Manifest

Die Prinzipien des agilen Manifests sind:

- Ergebnisorientierung
- Einfachheit
- Kontinuierliche Auslieferung von Software in kurzen Zeitspannen
- Tägliche Zusammenarbeit zwischen Fachexperten und Entwicklern
- Direkte Informationsübermittlung



Agile Manifest

Die Prinzipien des agilen Manifests sind:

- Motivierendes und unterstützendes Umfeld
- Einhaltung einer einheitlichen Arbeitsgeschwindigkeit
- Funktionsfähigkeit als Kennzahl des Fortschritts der Software
- Veränderungen sind stets gewünscht
- Selbstorganisation des Teams
- Selbstreflexion der Handlungen

Merkmale von Projekten

- Klare Zielsetzung
- Einmaligkeit
- Interdisziplinarität
- Restriktionen
 - Zeitliche Restriktionen
 - Monetäre Restriktionen
 - Personelle Restriktionen
 - ...

Projektmanagement

Projektmanagement hat die Verantwortung für Planung, Koordination, Steuerung und Überwachung des Projekts.

Sichten des Projektmanagements:

- Funktionale Sicht
- Institutionelle Sicht
- Instrumentelle Sicht

Das Projektmanagement befindet sich in einem magischen Dreieck.

Projektmanagement

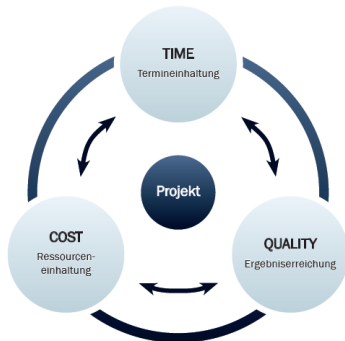
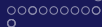


Abbildung: Magisches Dreieck

Quelle: <http://www.projektmanagementhandbuch.de/projektinitiierung/zieldefinition/>



Projektmanagement

Die Phasen eines Projekts sind die folgenden:

- Analysephase
- Entwurfsphase
- Realisierungsphase
- Einführungsphase

Der Ablauf der Phasen eines Projekts werden durch Vorgehensmodelle beschrieben.



Projektmanagement

Die Merkmale der agilen Vorgehensmodelle sind die folgenden:

- Zyklischer Durchlauf der Phasen
- Einhaltung der Prinzipien des agilen Manifests
- Balance zwischen Struktur und Flexibilität
- Projektmanagementansatz zur Förderung der kooperativen Arbeit

Abgrenzung zu anderen Konzepten der Softwareentwicklung

Wasserfallmodell

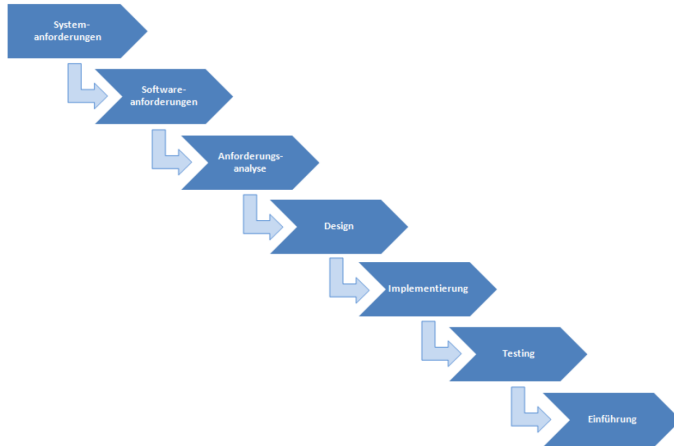


Abbildung: Wasserfallmodell Quelle: Mark Harwardt



Wasserfallmodell

Kritik

- Auftraggeber wird nur am Anfang und am Ende involviert
- Erfordert eine umfassende Spezifikation am Anfang des Projekts
- Änderungen sind weitestgehend ausgeschlossen
- Späte Identifizierung von Fehlern
- Spätes Feedback des Auftraggebers

Allgemeines

- Entwickelt von Kent Beck im Jahr 2000
- Extreme Programming definiert:
 - 5 Werte
 - 14 Prinzipien
 - 13 Primärpraktiken
 - 11 Folgepraktiken

5 Werte des XP

- 1 Kommunikation
- 2 Einfachheit
- 3 Rückkopplung
- 4 Mut
- 5 Respekt

14 Prinzipien I

- 1 Menschlichkeit
- 2 Wirtschaftlichkeit
- 3 Gegenseitiger Vorteil
- 4 Selbstähnlichkeit
- 5 Verbesserung
- 6 Mannigfaltigkeit
- 7 Reflexion

14 Prinzipien II

- 8 Fluss
- 9 Gelegenheit
- 10 Redundanz
- 11 Fehlschlag
- 12 Qualität
- 13 Babyschritte
- 14 Akzeptieren von Verantwortlichkeiten

Phasen

Alle Phasen werden in Iterationen ausgeführt.

- Erzeugung von Inkrementen
- Iterationsdauer von ca. 2 Wochen
- Keine explizite Reihenfolge des Phasenablaufs



Phasen

Planung

- Schätzungen der Kosten und Risiken der Komponenten
- Beteiligung des gesamten Teams
- Komponentenauswahl des Auftraggebers
- Bestimmung von Akzeptanztests der Komponente



Phasen

Design

- Berücksichtigung nur von Aspekten der aktuellen Iteration
- Einhaltung der Tests

Kodieren

- Implementierung
- Integration des Auftraggebers

Testen

- Akzeptanztests
- Erstellung weiterer Tests



Phasen

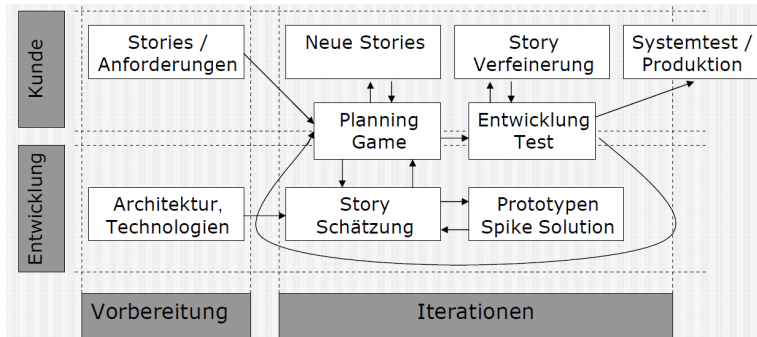


Abbildung: Ablauf des XP

Quelle: *eXtreme Programming (XP)* (Hermann Götz & Sergij Paholchak)

Rollen

Auftraggeber (auch als Kunde bezeichnet)

- In räumlicher Nähe der Entwickler
- Klärung geschäftlicher und fachlicher Fragestellungen

Entwickler

- Entwicklung der Software
- Kooperation mit dem Auftraggeber
- Entwicklerteams bis zu 15 Personen

XP-Coach

- Klärung methodischer und koordinativer Fragestellungen
- Anreizschaffung für Entscheidungsprozesse



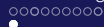
Primärpraktiken I

Die Primärpraktiken sind die folgenden:

- 1 Räumliche Zusammenarbeit
- 2 Komplettes Team
- 3 Informative Arbeitsumgebung
- 4 Energiegeladene Arbeit
- 5 Pair-Programming
- 6 User Stories

Primärpraktiken II

- 7 Wochenzyklus
- 8 Quartalszyklus
- 9 Freiraum
- 10 Zehn-Minuten-Build
- 11 Kontinuierliche Integration
- 12 Testgetriebene Entwicklung
- 13 Inkrementeller Entwurf



Weitere Verfahren I

Weitere agile Verfahren:

- ActiF
- Adaptive Software Development (ASD)
- Agile Enterprise (ehemals X Breed)
- Agile Model Driven Development (AMDD)
- Behavior Driven Development (BDD)
- Crystal
- Design Driven Development (D3)
- Dynamic System Development Method (DSDM)
- Eclipse Way Process

Weitere Verfahren II

- Evolutionary Process For Integrating Cots-Based Systems (EPIC)
- Evolutionary Project Management & Product Development (EVO)
- Feature Driven Development (FDD)
- Iconix
- Internet-Speed Development
- Lean Software Development (LSD)
- Microsoft Solutions Framework For Agile Software Development (MSF4ASD)
- Mobile-D
- Rapid Application Development (RAD)

Weitere Verfahren III

- Scrum
- Test Driven Development (TDD)
- Unified Process (UP)
- Agile Unified Process (AUP)
- Essential Unified Process (EssUP)
- Open Unified Process (OpenUP)
- Usability Driven Development (UDD)

Quelle: <http://www.computerwoche.de/a/agile-methoden-im-vergleich,2352712>

Chancen

■ Verbesserung der Kommunikation

[Quelle: *Agil-systemisches Softwareprojektmanagement* (Tobias Trepper)]

■ Identifizierung der Spezifikation

[Quelle: *Extreme Programming: Vorgehensmodell zur Software-Entwicklung bei //SEIBERT/MEDIA*
(Martin Seibert)]

■ Sofortige Berücksichtigung von neuen und geänderten Anforderungen

[Quelle: *eXtreme Programming* (Benedikt Franz)]

■ Software entspricht den Erwartungen

[Quelle: *Agile Softwareentwicklung im Vergleich zu traditionellen Softwareentwicklungsprozessen*
(E-Business Management)]



Chancen

■ Vermeidung von überflüssiger und veralteter Dokumentation

[Quelle: *Agil-systemisches Softwareprojektmanagement* (Tobias Trepper)]

■ Schnelle Auslieferung der Software

[Quelle: *Agile Softwareentwicklung im Vergleich zu traditionellen Softwareentwicklungsprozessen*

(E-Business Management)]

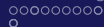
■ Frühzeitige Identifikation von Risiken

[Quelle: *eXtreme Programming (XP)* ((Götz & Paholchak)]

■ Identifikation der Entwickler mit dem Projekt

[Quelle: *Agile Softwareentwicklung im Vergleich zu traditionellen Softwareentwicklungsprozessen*

(E-Business Management)]



Chancen

■ Wissensweitergabe im Team

[Quelle: *eXtreme Programming (XP)* ((Götz & Paholchak))]

■ Generierung des größtmöglichen Wertes der Software

[Quelle: *eXtreme Programming (XP)* ((Götz & Paholchak))]

■ Frühzeitige Inbetriebnahme der Software

[Quelle: *eXtreme Programming (XP)* (it-agile GmbH)]



Risiken

■ Erfordernis einer optimalen personellen Besetzung

[Quelle: *Agil-systemisches Softwareprojektmanagement* (Tobias Trepper)]

■ Willensmangel des Auftraggebers

[Quelle: *Extreme Programming: Vorgehensmodell zur Software-Entwicklung bei //SEIBERT/MEDIA*

(Martin Seibert)]

■ Zwischenmenschliche Probleme

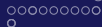
[Quelle: *Agil-systemisches Softwareprojektmanagement* (Tobias Trepper)]

■ Mangelnde Kontaktpflege der Entwickler mit dem Auftraggeber

[Quelle: *Agil-systemisches Softwareprojektmanagement* (Tobias Trepper)]

Risiken

- Starke Abhängigkeit der Ergebnisse von den Entwicklern
[Quelle: *Entwicklung: Agile - das Aus für Wasserfall?* (Computerwoche)]
- Folgen einer Mitarbeiterfluktuation
[Quelle: *Agil-systemisches Softwareprojektmanagement* (Tobias Trepper)]
- Große und verteilte Teams in XP kaum umsetzbar
[Quelle: *Agil-systemisches Softwareprojektmanagement* (Tobias Trepper)]
- Schlechte Übertragbarkeit auf große und sicherheitskritische Projekte
[Quelle: *Entwicklung: Agile - das Aus für Wasserfall?* (Computerwoche)]



Risiken

■ Schwierige Überwachung des Projekts

[Quelle: *Agil-systemisches Softwareprojektmanagement* (Tobias Trepper)]

■ Problematische Festsetzung von Inkrementen

[Quelle: *Agil-systemisches Softwareprojektmanagement* (Tobias Trepper)]

■ Neigung zur zu schnellen Fertigstellung

[Quelle: *Extreme Programming: Vorgehensmodell zur Software-Entwicklung bei //SEIBERT/MEDIA*

(Martin Seibert)]

■ Planung des Gesamtpreises

[Quelle: *Agil-systemisches Softwareprojektmanagement* (Tobias Trepper)]

Übersicht

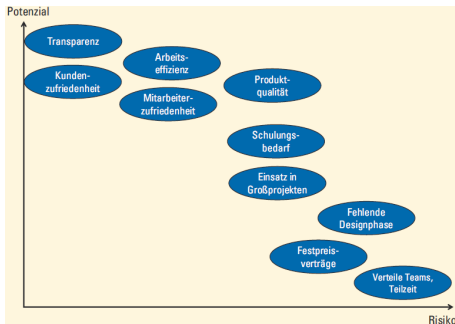


Abbildung: Übersicht der Chancen & Risiken

Quelle: <http://www.pmaktuell.org/uploads/PMAktuell-201301/PMAktuell-201301-033-Public.pdf>

Zusammenfassung

Das agile Manifest bildet die Grundlage für die agilen Vorgehensmodelle, wie dem Extreme Programming. In XP werden dazu Werte, Prinzipien und Praktiken definiert, die die Vorgehensweise explizit festlegt.

Es bietet Chancen, wie den Umgang mit Auftraggebern, die keine konkrete Vorstellung von der Software haben, Flexibilität, die Änderungen erfordern bis hin zur Identifikation der Entwickler mit dem Projekt.

Dem stehen die Risiken, der Erfordernis der hochqualifizierten Entwicklern, der zwischenmenschlichen Probleme und der schweren Festsetzung von Inkrementen, gegenüber.



Literaturverzeichnis I



Computerwoche.

Agile Methoden im Vergleich.

<http://www.computerwoche.de/a/agile-methoden-im-vergleich,2352712>, 2013.
[Online; Stand 14. Juni 2014].



Computerwoche.

Entwicklung: Agile - das Aus für Wasserfall?

<http://www.computerwoche.de/i/detail/artikel/2352228/1/943975/d2e274-media/>, 2013.
[Online; Stand 21. Juni 2014].

Literaturverzeichnis II



Benedikt Franz.

eXtreme Programming.

[http:](http://dme.rwth-aachen.de/en/system/files/file_upload/course/12/proseminar-methoden-und-werkzeuge/cameraready-extremeprogramming.pdf)

[//dme.rwth-aachen.de/en/system/files/file_upload/
course/12/proseminar-methoden-und-werkzeuge/
cameraready-extremeprogramming.pdf.](http://dme.rwth-aachen.de/en/system/files/file_upload/course/12/proseminar-methoden-und-werkzeuge/cameraready-extremeprogramming.pdf)

[Online; Stand 21. Juni 2014].

Literaturverzeichnis III



Hermann Götz; Sergij Paholchak.

eXtreme Programming (XP).

http://www3.informatik.uni-wuerzburg.de/courses/vorl_06_ss/projman/daten/referate_new/eXtreme_Programming__XP_.pdf.

[Online; Stand 21. Juni 2014].



Eckhart Hanser.

Agile Prozesse: Von XP über Scrum bis MAP.

Springer-Verlag, 2010.

Literaturverzeichnis IV



Mark Harwardt.

Wasserfallmodell versus Scrum.

<http://www.fernuni-hagen.de/imperia/md/content/ps/masterarbeit-harwardt.pdf>.

[Online; Stand 12. Juni 2014].



Peter Hruschka; Chris Rupp; Starke Gernot.

Agility kompakt.

Spektrum Akademischer Verlag, 2009.

Literaturverzeichnis V



Martin Seibert.

Extreme Programming: Vorgehensmodell zur Software-Entwicklung bei //SEIBERT/MEDIA.

<http://blog.seibert-media.net/blog/2005/05/01/extreme-programming-vorgehensmodell-zur-software-entwicklung-2005>.

[Online; Stand 21. Juni 2014].



Tobias Trepper.

Agil-systemisches Softwareprojektmanagement.
Springer Gabler Verlag, 2012.

Literaturverzeichnis VI



it-agile GmbH.

eXtreme Programming (XP).

[http://www.it-agile.de/wissen/methoden/
extreme-programming/](http://www.it-agile.de/wissen/methoden/extreme-programming/).

[Online; Stand 21. Juni 2014].