



Die Rolle der Anforderungen

Alex Bögli

Software Engineering Seminar



Inhalt

- Rolle in klassischen Methoden
- Rolle in agilen Methoden
 - Extreme Programming
 - Extreme Requirements
 - Scrum
- Fazit



Klassische Methoden

■ Vertreter

- „Urväter“ wie Wasserfall- oder Spiralmodell
- Rational Unified Process
- V-Modell 97

■ Ziele

- Planbarkeit der Entwicklung
- Entwicklung nach Plan



Klassische Methoden

- Hauptproblem
 - Zeitliche Differenz resp. fehlendes Feedback
- Lösungsansatz
 - Vorgabe von Tätigkeiten
 - Klar definierte Ergebnisse
 - Gegliedert in Phasen
 - Genaue Durchführung und Dokumentation



Requirements Engineering

- Genaue Analyse und Dokumentation
- Tätigkeiten
 - Requirements elicitation
 - Requirements analysis and negotiation
 - Requirements documentation
 - Requirements validation



Techniken

- Interviews
- Szenarien
- Prototyping
- Teilnehmerüberwachung
- ...



Rolle der Anforderungen

- Stabile Anforderungen als Voraussetzung
- Wenn stabil, dann
 - ⇒ fehlendes Feedback keine Problem
 - ⇒ Anforderungen auch als Vertragsgrundlage verwendbar



Agile Methoden

■ Ziele

- Anpassungsfähigkeit
- Personen-Orientierung

■ Lösungsansatz

- Schnelles Feedback durch kurze Zyklen
- Vermeidung von Bürokratie
- Mehr Kommunikation



Manifest

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- **Customer collaboration over contract negotiation**
- Responding to change over following a plan



Extreme Programming

- Verschiedene „Best Practices“
- Stärkung der Zusammenarbeit
- Kontinuierliche Tests
- Wenig Dokumentation



User Stories

- Beschreiben eine Funktionalität (Feature)
- Vom Kunden geschrieben
- Kurz (passen auf A6-Kärtchen)
- Grundlage für Release-Planung

„The story represents a concept and is not a detailed specification. *A user story is nothing more than an agreement that the customer and developers will talk together about a feature.*“ (Kent Beck)



Ablauf

- Story-Schreiben üben
- Erster Satz an Stories schreiben
(Hauptanforderungen)
- Umsetzen der Stories, gleichzeitig Stories
hinzufügen, ändern, entfernen
- Story-Kärtchen vernichten



Beurteilung

- Erfrischend einfach und schlicht
- Anforderungen keine Vertragsgrundlage
- Kunde vor Ort unabdingbar
- Keine Unterstützung/Techniken zur Anforderungsgewinnung



Extreme Requirements

- Weiterentwicklung von XP
- Adressiert Probleme im Zusammenhang mit Anforderungen



Problem 1

Repräsentation aller Stakeholders durch einen Kundenvertreter.

Daher:

Stories mit Szenarien ergänzen und von anderen Kundenvertretern validieren lassen



Problem 2

Fehlende Beachtung nichtfunktionaler Anforderungen

Daher:

Möglichst früh ebenfalls definieren und zusammen mit funktionalen Anforderungen festhalten



Problem 3

Fehlende explizite Verbindung zwischen Stories, Tasks und Code

Daher:

Führen eines Glossars mit Begriffen aus der relevanten Geschäftswelt und konsequente Verwendung dieser Begriffe



Problem 4

Kein definierter Prozess zur Definition
funktionaler Tests

Daher:

Stories mit Szenarien ergänzen und daraus
Testfälle ableiten, insbesondere auch die
Ausnahmefälle



Problem 5

Kein definierter Prozess zur Gewinnung von Stories und Tasks

Daher:

Definition eines solchen Prozesses, auch zur Gewinnung der Szenarien und inklusive Verifizierung und Validierung



Beurteilung

- Sehr gute Übersicht der Probleme von XP
- Lösungen zu stark aus klassischen Methoden entnommen

⇒ Agilität geht verloren



Scrum

- Leichtgewichtiger Management-Prozess
- Iterative Entwicklung (Sprints)
- Kleine Teams mit Scrum Master
- Tägliche Meetings



Product Backlog

- Enthält Anforderungen an System
- Nach Prioritäten geordnet
 - Hohe Priorität ⇒ detailliert
 - Niedrige Priorität ⇒ weniger detailliert
- Allen zugänglich
- Ausschliesslich vom Product Owner verwaltet



Product Owner

- Nimmt Anforderungen entgegen
- Entscheidet über Aufnahme in Product Backlog
- Priorisiert Einträge im Product Backlog
- Schätzt Aufwand zusammen mit Team



Ablauf

- Auswahl und Fixierung einer Teilmenge aus Product Backlog
- Unterteilung in Backlog Items
- Umsetzung in einer Iteration
- Begutachtung des Erreichten und Planung des weiteren Vorgehens



Beurteilung

- Stabilität während Sprint
- Keine „Versionierungsprobleme“

- Anforderungen keine Vertragsgrundlage
- Keine Unterstützung/Techniken zur Anforderungsgewinnung



Fazit

- Agile Methoden sehr ähnlich
 - Zentralisiert Liste von Anforderungen
 - Entwickler stark entlastet
 - Kleine Iterationen ermöglichen Korrekturen
- Gleiche Mängel
 - Keine Unterstützung bei Gewinnung
 - Keine Beachtung nichtfunktionaler Anforderungen



Fazit

- Klar unterschiedliche Rollen in klassischen und agilen Methoden

Klassisch	Agil
<ul style="list-style-type: none">■ Grundlage für Entwicklung■ Stabile Basis■ Vertragsinhalt	<ul style="list-style-type: none">■ Grundlage für Entwicklung■ Steuert Entwicklung

⇒ Art der Anforderungen entscheidet über Anwendbarkeit des Prozesses