

# Softwareanforderungsanalyse

## Evolution von Anforderungen

Burkhardt Renz

THM, Fachbereich MNI

Wintersemester 2018/19

# Evolution von Anforderungen

- Anforderungen **entwickeln** sich fort
  - Neue Märkte, neue Kunden, neue Geschäftsprozesse
  - Technischer Fortschritt
  - Änderung rechtlicher Bedingungen
- Anforderungen **ändern** sich während der Entwicklung
  - Verständnis des Anwendungsgebiets wächst
  - Neue Möglichkeiten werden entdeckt
  - Mängel werden sichtbar

Also braucht man

- **Antizipation** von Änderungen
- **Konfigurationsmanagement**
- **Rückverfolgbarkeit von Anforderungen**

# Übersicht

- Versionen und Varianten
- Antizipation von Änderungen
- Änderungsprozess
- Rückverfolgbarkeit (Traceability)
- Monitoring und dynamische Anpassung

# Versionen und Varianten

**Evolution** der Anforderungen hat zwei Dimensionen:

## Versionen

Eine neue **Version** ergibt sich bei der Verbesserung oder Veränderung von Anforderungen für ein bestimmtes System oder Feature eines Systems.

Änderung in der Dimension der **Zeit**.

## Varianten

Eine **Variante** ergibt sich dadurch, dass auf einer gemeinsamen Basis verschiedene Ausprägungen eines Systems mit unterschiedlichen Anforderungen im Detail gebildet werden.

Änderung in der Dimension des Design-**Raums**

- typisch für **Softwareproduktlinien**.

# Übersicht

- Versionen und Varianten
- Antizipation von Änderungen
- Änderungsprozess
- Rückverfolgbarkeit (Traceability)
- Monitoring und dynamische Anpassung

# Antizipation von Änderungen

- Wahrscheinlichkeit von Änderungen bei der Spezifikation von Anforderungen berücksichtigen
- Alternative Möglichkeiten dokumentieren
- Gegebenheiten des Anwendungsgebiets bzgl. Änderungen einbeziehen
- Features in Bezug auf ihre Stabilität gruppieren
- Änderungsnotwendigkeit kann selbst eine Anforderung sein: adaptierbare Software

# Übersicht

- Versionen und Varianten
- Antizipation von Änderungen
- **Änderungsprozess**
- Rückverfolgbarkeit (Traceability)
- Monitoring und dynamische Anpassung

# Änderungsprozess

- **Geregeltes** Verfahren für die Änderung von Anforderungen
- Typischerweise durch einen **Lenkungsausschuss**, auch *Change Board* genannt
- Ablauf, typischerweise iterativ:
  - Änderungsantrag
  - Vorprüfung: warum? notwendig?
  - Analyse der Auswirkungen
  - Änderungsvorschlag
  - Beschluss



# Versionierung

- Man bildet **Konfigurationen** von Anforderungen, die logisch zusammenhängen, konsistent sind, eindeutig identifizierbar sind
- Eine stabile Konfiguration ist eine **Baseline**, oft die Spezifikation für ein Release eines Produkts oder einer Produktlinie
- Werkzeugsunterstützung durch **Versionsverwaltung** oder spezielle Tools für das Requirements Engineering wie IBM Rational DOORS (Dynamic Object Oriented Requirements System)

# Übersicht

- Versionen und Varianten
- Antizipation von Änderungen
- Änderungsprozess
- Rückverfolgbarkeit (Traceability)
- Monitoring und dynamische Anpassung

# Rückverfolgbarkeit

- Eine Anforderung ist (rück-)verfolgbar, wenn man weiß, **woher** sie kommt, **warum** sie besteht, **was** mit ihr **wo** erreicht wird und **wie** sie umgesetzt wird.
- **Traceability** besteht im Identifizieren, Dokumentieren und Wiederfinden von Gründen und Auswirkungen von Anforderungen
- benötigt um **Umsetzung** von Anforderungen zu belegen
- benötigt um **Folgen von Änderungen** zu überblicken

# Typen von Traceability

- **Rückwärts** – *pre-requirements-specification traceability*  
Woher kommen die Anforderungen, was begründet sie?
- **Vorwärts** – *post-requirements-specification traceability*  
Wo wird die Anforderung im Entwurf und der Implementierung umgesetzt?
- **Abhängigkeiten** von Anforderungen  
Wie hängen sie voneinander ab? z.B. Verfeinerung, Verwendung etc

# Repräsentation von Verfolgbarkeits-Links

- **Hyperlinks** zwischen Dokumenten
- **Matrix** für Abhängigkeiten zwischen Anforderungen
- **Graph** der Abhängigkeiten
- **Feature-Diagramme** der Aufgliederung und Abhängigkeiten von Features (gut geeignet für die Dimension *Varianten*)

Werkzeuge beispielhaft:

- Itemis Yakindu Traceability <http://www.yakindu.de>
- Feature IDE <https://featureide.github.io>

Das **Zielemodell**, das wir später betrachten, enthält in seiner Struktur Erfüllbarkeitsargumente

→ Traceability gewissermaßen eingebaut

# Übersicht

- Versionen und Varianten
- Antizipation von Änderungen
- Änderungsprozess
- Rückverfolgbarkeit (Traceability)
- **Monitoring und dynamische Anpassung**

# Monitoring und dynamische Anpassung

- Motivation: Laufzeitumgebung ist nicht vollständig vorhersehbar
- Antizipierte Variabilität bzgl. dieser Bedingungen
- Lösung: Anpassbares System
  - Adaption zur Installationszeit
  - Adaption zur Laufzeit
- Vorkehrungen
  - Monitore und Anpassbarkeit werden Teil der Anforderungen
  - Metalevel-Architektur in Entwurf und Entwicklung
  - Laufzeitüberprüfung und Adaption des Systems

# Fazit

- **Evolution** im Anwendungsgebiet und doch Konsistenz, Vollständigkeit und Adäquatheit der Anforderungen
- Dazu muss man Änderungen **antizipieren**
- und einen geregelten **Änderungsprozess** haben
- und Anforderungen **rückverfolgbar** machen
- In manchen Situation ist eine **Meta-Level-Architektur** zur Unterstützung von Evolution und Variabilität erforderlich (mehr darüber und über andere Mechanismen in der Literatur über **Softwareproduktlinien**)