

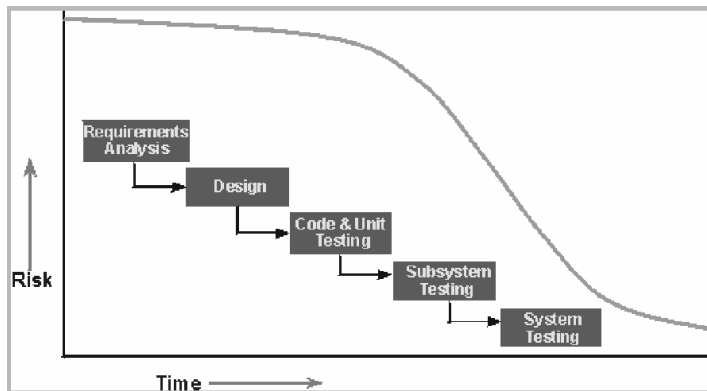
Requirements

„Requirements Engineering befasst sich mit den Methoden und Werkzeugen zur Erhebung der Anforderungen an Softwaresysteme und stellt damit eine zentrale Aktivität in der Softwareentwicklung dar.“

Eine Anforderung sollte...

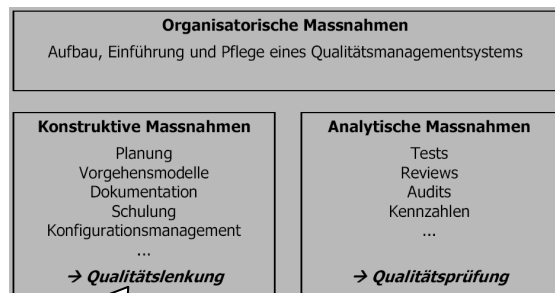
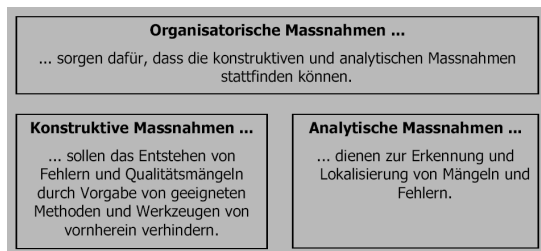
atomar, identifizierbar, vollständig, korrekt, machbar, notwendig, priorisiert, eindeutig, verständlich, nachprüfbar, rück- und vorwärtsverfolgbar... sein.

Softwareentwicklungsprozess



- Strukturierter Arbeitsablauf
- Vereinheitlichung
- Dokumentation
- Qualitätssicherung
- Nachvollziehbarkeit

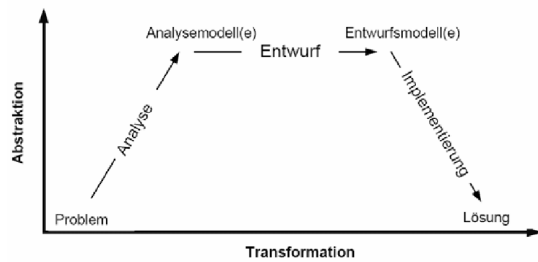
Entwicklungsprozess und Softwarequalität



= Qualitätssicherung;
Requirements
Engineering

Dem
Entwicklungsprozess
nachgelagert.

Vorgehensmodelle



⇒ Gliederung in Teilaktivitäten/Phasen
(Inputs, Outputs, Beteiligte, Methoden)

Probleme

- Bestimmung der Gesamtaufgabe (Kommunikation, Komplexität, Dynamik)
- Unterteilung in sinnvolle Teilaktivitäten (Granularität, Überschneidungen)

Nutzen von Vorgehensmodelle

- Bestimmung Tätigkeiten:
Aufgaben je Teilaktivität erkennen und definieren: Ziele, Alternativen, Restriktionen (Zeit, Ressourcen), Risikobewertung (Falsche Annahmen, Missverständnisse in der Kommunikation)
- Zusicherung Konsistenz
- Schaffung von Kontrollpunkten

Grundphasen der Vorgehensmodelle

Analyse und Planung

- ⇒ Pflichtenheft mit Zieldefinition und Anforderungskatalog
- Planung des Projekts

Design

- ⇒ Rahmenkonzept und Feinkonzept (Organisations-, Daten-, Steuerungs-, Funktionssicht)
- Projektstrukturplan
- Layoutkonzept (Accessibility)
- technisches Konzept
- Funktionsabläufe (Storyboard)

Implementierung

- ⇒ Eigentliche Umsetzung
- Aufgabenplanung, Zeitplanung, Ressourcenplanung

Dokumentation

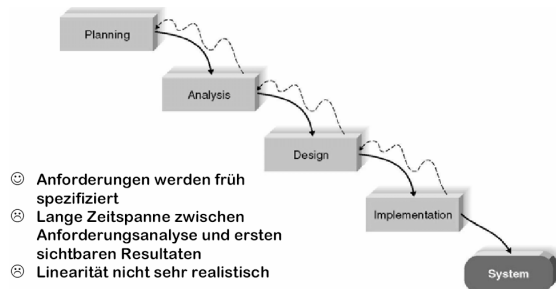
- ⇒ Aufbau des Systems, Benutzerhandbuch

Test und Wartung

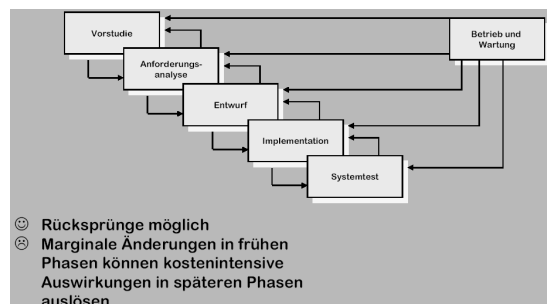
- ⇒ Analysebericht
- Anforderungen erfüllt? Testfälle korrekt abgearbeitet? Stabil? Antwortzeiten?

Vorgehensmodelle

Wasserfallmodell

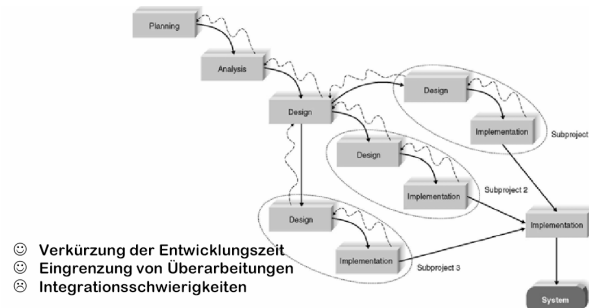


- strikt sequentiell (aber zeiteffizient)
- starr (keine Abweichungen vorgesehen)
- jede Projektphase endet mit einem Validierungsprozess (strikte Phasentrennung)
- Rücksprung nur erlaubt, wenn Ergebnisse der vorangehenden Phase eindeutig fehlerhaft sind.
- Praxis: Oft nur zu Beginn eines Projekt eingesetzt (Planungsphase); nicht bei neuartigen Projekten
- unterstützt kompakte, zielstrebige Entwicklungen



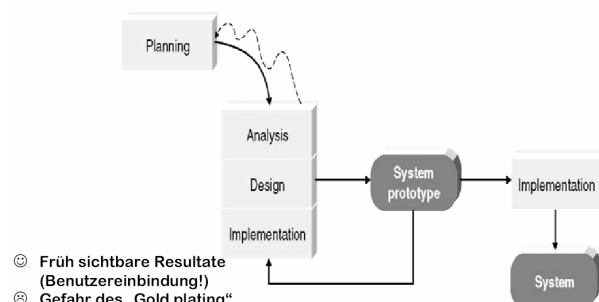
- Erweiterung
Interaktiv, zyklisch (Rücksprünge erlaubt)

Parallelisierte Entwicklung



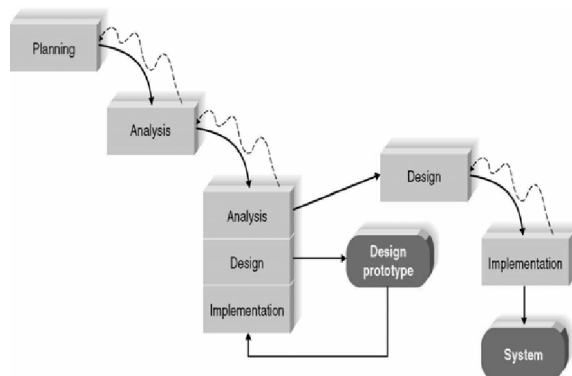
- Leitplanken des Gesamtsystems, anschliessend modulare Entwicklung der Teilsysteme.
- Teilssystementwicklung gemäss Wasserfall, Ergebnisse schon früh erkennbar.
- Integration der Teilsysteme (schwierig!)

Prototyping



- Entwicklung Prototyp aus groben Anforderungsspezifikationen
- schrittweise Verfeinerung (Lernprozess)
⇒ Ergebnis entspricht Anforderungen
⇒ Teurer Lernprozess
- Oft bei „Baukastensystemen“ eingesetzt
- keine stabile Struktur und schwer wartbar

Wegwerf-Prototyping



- Prototyp dient der Klärung der Anforderungen
- Extrem kostspielig (Doppelentwicklung), dafür kleineres Risiko, ein unbrauchbares System entwickelt zu haben.

Spiralmodell



- Integration verschiedener Ansätze (Prototyping, evolutionäre Entwicklung, Validierungsschritte, Managementkonzepte etc.)
- Beispiel: detaillierter Prototyp
 Prototyp 1: Modellierung
 Prototyp 2: Oberfläche GUI
 Prototyp 3: Lauffähiger Typ
 Prototyp 4: Produktiver Typ
- Start: Mittelpunkt
- radiale Dimension: kumulative Kosten
- Winkel: Fortschritt des Projekts
- Für jeden Zyklus werden die gleichen Phasen durchlaufen.