



# Vorgehensmodelle der Software-Entwicklung

## Informationswirtschaft II

Wolfgang H. Janko, Stefan Koch, Michael Hahsler

# Inhaltsverzeichnis

1. Begriffsdefinition
2. Nutzen von Vorgehensmodellen
3. Anwendung und Customizing von Vorgehensmodellen
4. Grundlegende Schritte eines Projektes
5. Paradigmen von Vorgehensmodellen
  1. Wasserfall-Modell
  2. Evolutionäre Ansätze
6. Qualitätssicherung

# Begriffsdefinition

## Vorgehensmodelle

- Aufbau der Software-Entwicklung
- Gliederung mit Meilensteinen
- enthalten, bzw. schlagen verschiedene Methoden vor

# Nutzen von Vorgehensmodellen

1. **Planungssicherheit:** vergleichbare Produkte vergleichbar erstellen
2. **Effizienz:** nichts vergessen, keine Nachbereinigungen
3. **Kostenreduzierung:** Fehler schnell erkennen
4. **Qualität:** von Anfang an kontrolliert

# Anwendung und Customizing von Vorgehensmodellen

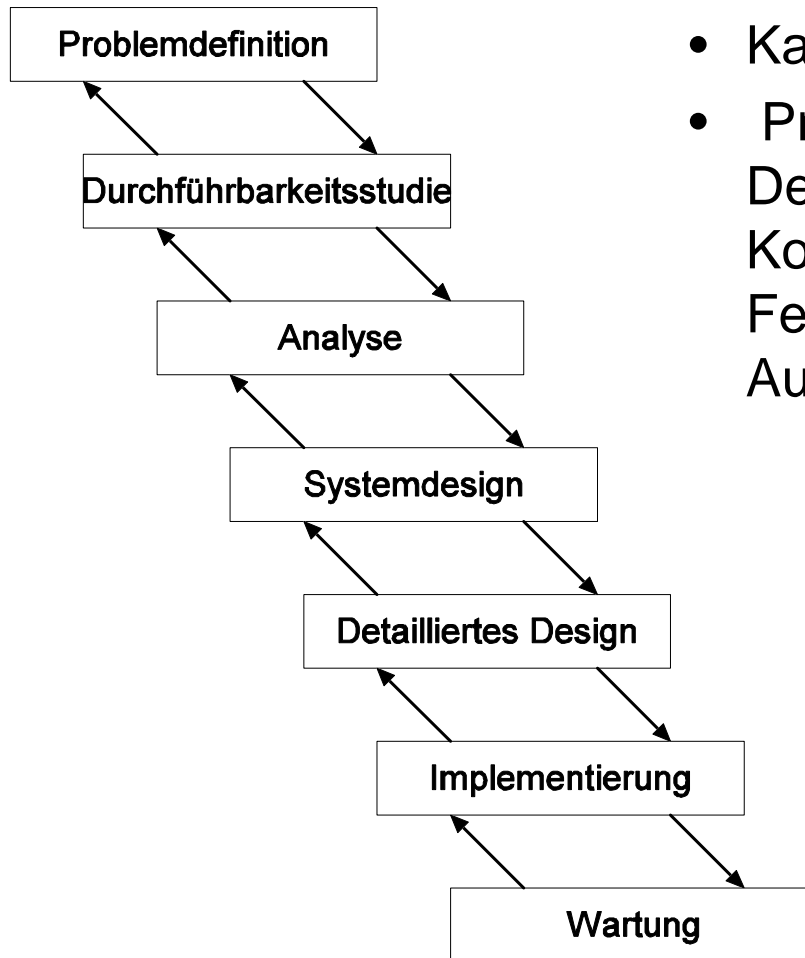
- Software-Entwicklung ist auch ein **Prozess** und kann und sollte modelliert werden (Software Process)
- damit auch für Organisationen und einzelne Projekte **anpassbar**, es sind nicht immer alle Schritte eines Vorgehensmodells nötig (Customizing)
- *der Software Process* wird in einer Organisation **iterativ** über mehrere Projekte **anpassen** (Software Process Improvement)

-> Vorgehensmodell als Referenzmodell

# Grundlegende Schritte eines Projektes

1. Analyse (was?)
2. Design (wie?)
3. Implementierung
4. (Wartung)

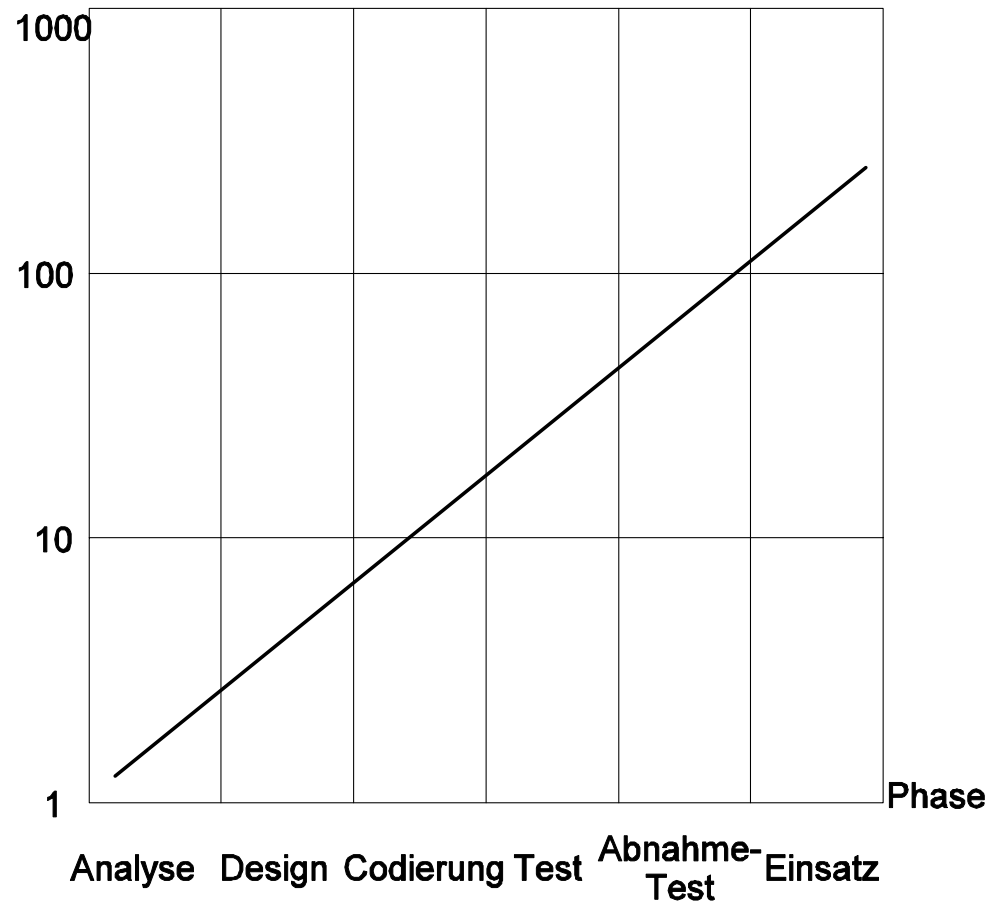
# Wasserfall-Modell



- Kaum Rückschritte in vorige Stufen
- Probleme, wenn ein Fehler im Design spät entdeckt wird (hohe Kosten der Behebung je später ein Fehler entdeckt wird, s.a. Aufwandsschätzung)

# Wasserfall-Modell: Korrekturkosten

Relative Kosten der Fehlerkorrektur



**Fehler im Einsatz: 100x teurer**

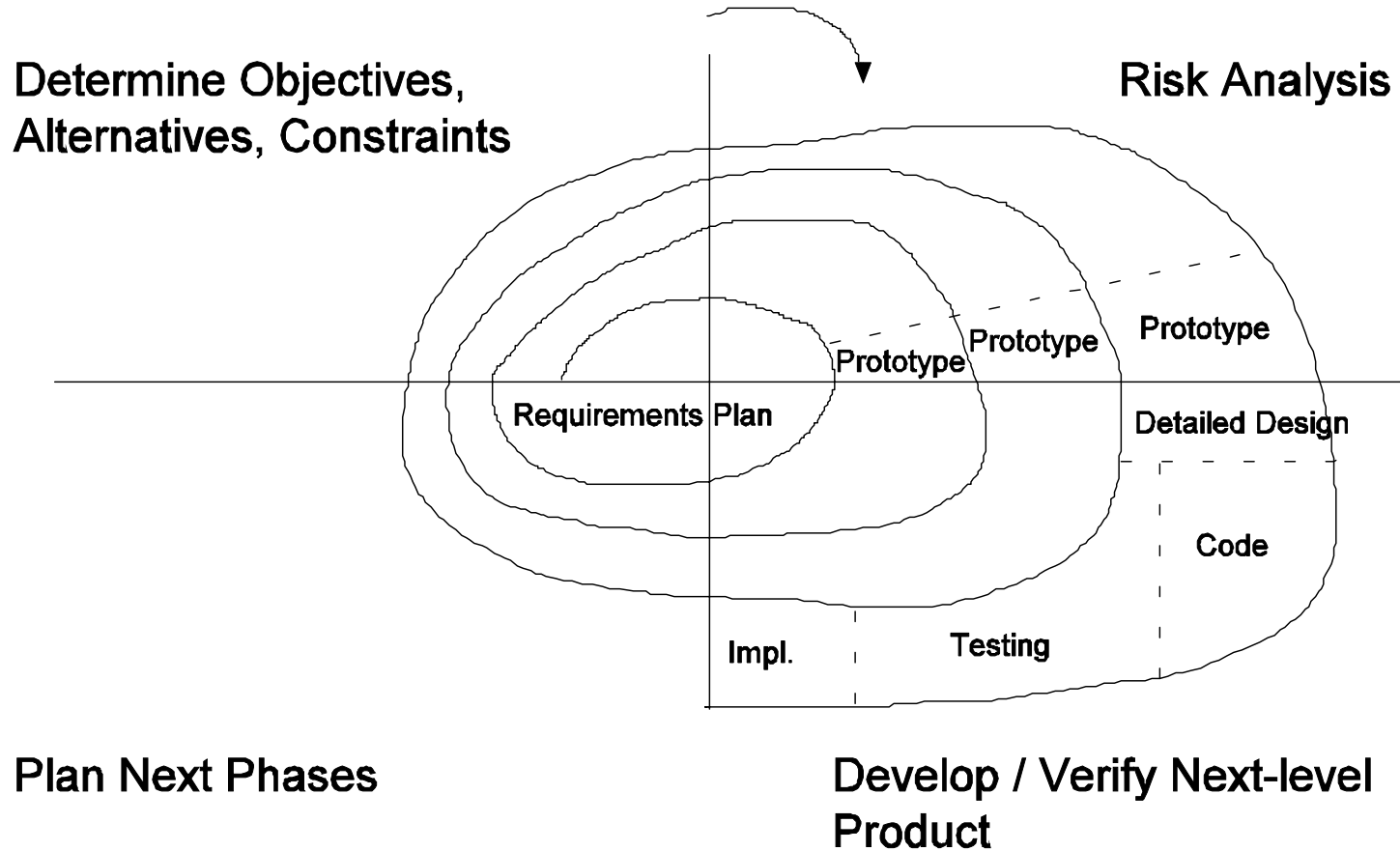
# Evolutionäre Ansätze

- Rückschritte in vorige Phasen
- Lernen
- Iteratives Vorgehen
- Prototyping

## -> Spiral-Modell

(Barry W. Boehm, 'A Spiral Model for Software Development and Enhancement', IEEE Computer, 21(5):61-72, May 1988.)

# Spiral-Modell



# Sicherung der Qualität des Softwareentwicklungsprozesses

- Beispiel: **CMM (Capability Maturity Model)**
  - Für Ausschreibungen vom US-Verteidigungsministerium entwickelt
  - definiert 5 Reifegrade (Maturity Levels) je nachdem, inwieweit Software Process unter Kontrolle ist und demzufolge die Qualität der erstellten Software gesichert ist
  - Dokumentation des Prozesses
  - Erhebung von Kennzahlen
  - Feedback in Prozessgestaltung
- Ähnliche Ansätze: Bootstrap, ISO 9001