



---

# Programmiermethodik

## Softwareentwicklung

### SS 2002

**Thomas Kühne**

[kuehne@informatik.tu-darmstadt.de](mailto:kuehne@informatik.tu-darmstadt.de)

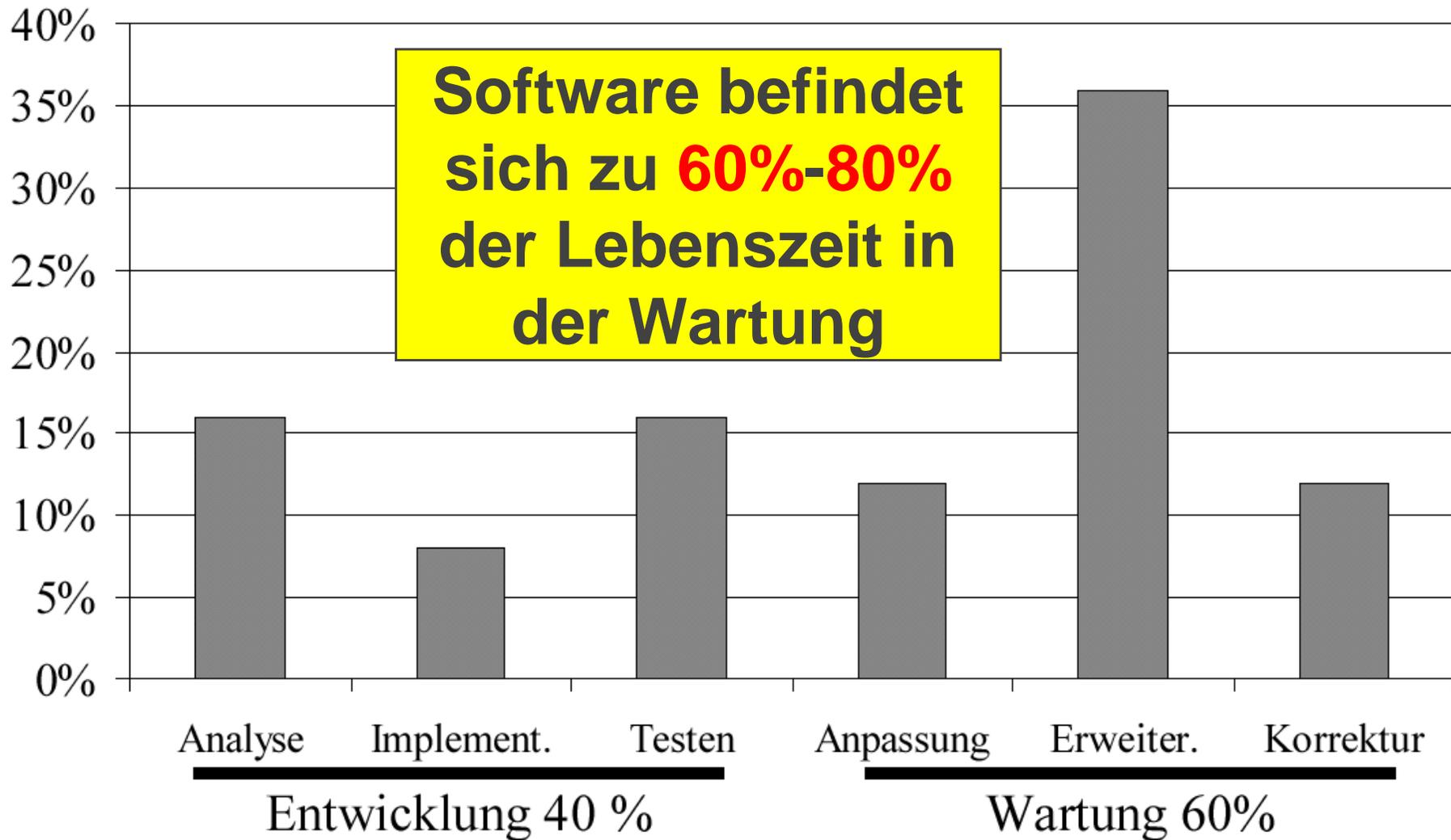
<http://www.informatik.uni-mannheim.de/informatik/softwaretechnik>



# Warum Entwicklungsmethodik?

---

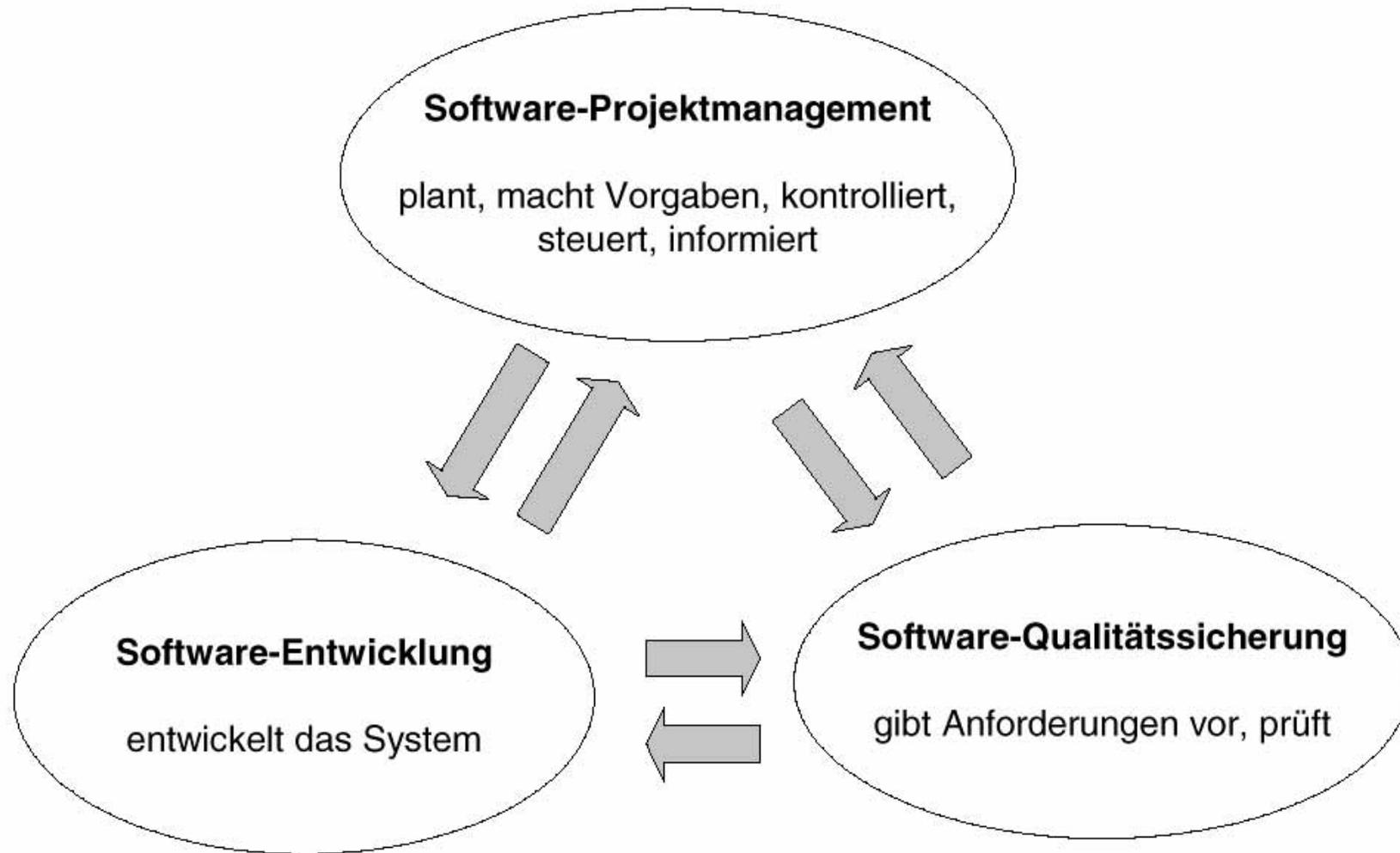
- Realisierung soll termingerecht und kostengünstig erreicht werden
  - » Bei großen kostenspieligen Entwicklungen ist es unabdingbar Aufwandsabschätzungen durchzuführen und Lösungsstrategien durchzuspielen
- Qualitätsansprüche, z.B., Ausbaufähigkeit
  - » Eine Realisierung, die gerade so funktioniert aber z.B., nicht mehr an veränderte Wünsche / Bedingungen angepaßt werden kann, hat ihre Bestimmung verfehlt



## Definition

- Allgemeiner Entwicklungsplan, der das generelle Vorgehen beim Entwickeln eines Software-Produkts festlegt
- Festlegung, welche Aktivitäten in welcher Reihenfolge von welchen Personen erledigt und welche Ergebnisse (Artefakte) dabei entstehen und wie diese überprüft werden

# SE-Teilmodelle





# Projektmanagement

---

- Terminplanung
  - » (wann welche Phasen)
- Personalplanung
  - » (Rollenzuteilung, Teambildung)
- Kostenplanung
  - » (Aufwand, Leistungsfähigkeit der Teams)
- Planung der Qualitätssicherung
  - » (welche Tests, wann)



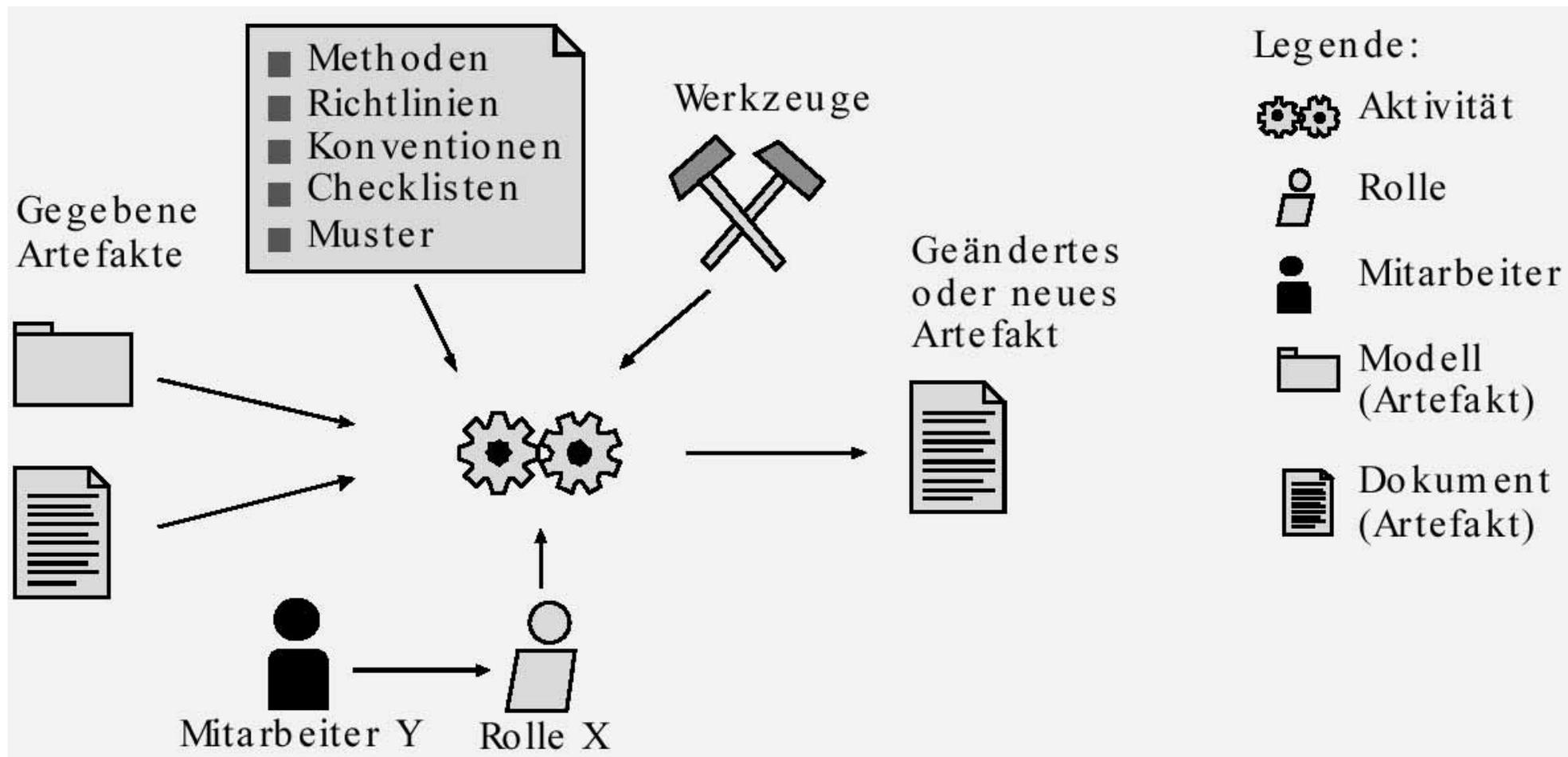
# Qualitätssicherung

---

- Korrektheit      Erwartungen erfüllen
- Robustheit      "Streß"-Situationen bestehen
- Bedienbarkeit      intuitiver, effizienter Umgang
- Effizienz      Geschwindigkeit & Kompaktheit
- Wartbarkeit      kostengünstige Änderungen
- Testbarkeit      Validierungsmöglichkeiten
- Wiederverwendbarkeit      Anwendung in anderen Kontexten
- Portierbarkeit      Plattformunabhängigkeit

**später mehr...**

## Durchführen einer Aktivität



- Artefakt

- » Ein greifbares Stück Information, das durch Mitarbeiter erzeugt, geändert und benutzt wird, wenn sie Aktivitäten ausführen
- » Kann ein Modell, ein Modellelement oder ein Dokument sein  
Beispiele: *Dokument*, z.B. Lastenheft, *Modell*, z.B. objektorientiertes Analysemodell, *Quellcode*, z.B. Java Programm.

- Software-Produkt
  - » Definierte Menge von Artefakten, insbesondere Realisierung (Code)
- Rolle
  - » Beschreibt die notwendigen Erfahrungen, Kenntnisse und Fähigkeiten, über die ein Mitarbeiter verfügen muss, um eine bestimmte Aktivität durchzuführen.



# Rollenverteilung (Beispiel)

---

- **1 Systemarchitekt**

- » Organisation des Teams
- » Entwurfsarchitektur
- » Betreuung
- » Überwachung

- **n Programmierer**

- » Feinentwurf
- » Programmierung der Module

- **1 Dokumentierer**

- » Spezifikationen erstellen und verwalten
- » Programmdokumentation
- » Benutzerhandbuch

- **1 Qualitätssicherer**

- » Code Inspektion
- » Testfälle erstellen
- » Tests durchführen und dokumentieren



# "Pair Programming"

---

## Programmieren in Zweiergruppen

- Unerfahrene lernen von Erfahrenen
  - » jedoch nicht "einer tut, der andere sieht zu"!
  - » gemeinsamer Dialog über Lösungen
- Gegenseitige Kontrolle
  - » Korrekturen durch den "inaktiven" Partner
  - » Rollenverteilung "Taktik" & "Strategie"
- Kommunikation des Entwurfs
  - » wechselnde Paarkombinationen

## Phasen (unvollständig)

- Planungsphase
  - » Zielvorstellung
- Analyse
  - » Verstehen des Problems
- Entwurf
  - » Konstruktion der Lösung
- Implementierung
  - » Realisierung

## Produkte

- ⇒ Lastenheft
- ⇒ Domänenmodell
- ⇒ Lösungsarchitektur
- ⇒ laufendes System



# Phasendefinition

---

- Notwendige Aktivitäten, um das Produkt weiterzuentwickeln
- Festlegungen pro Phase:
  - » Ziele der Phase
  - » Durchzuführende Aktivitäten
  - » Aktivitäten/Rollenzuordnung
  - » Zu erstellende Artefakte
  - » Zu beachtende Methoden, Richtlinien, Checklisten
  - » Meilenstein(e)
  - » Einzusetzende Werkzeuge und Sprachen.

## Aufgabe

- Definition des Auftrags
- Machbarkeitsstudie

## Relevante Artefakte

- Use Case Diagramme
  - » Anwendungsfälle

Grobes  
Pflichtenheft

## Meilenstein

- Lastenheft, Projektkalkulation und -plan



# Projektentscheidung

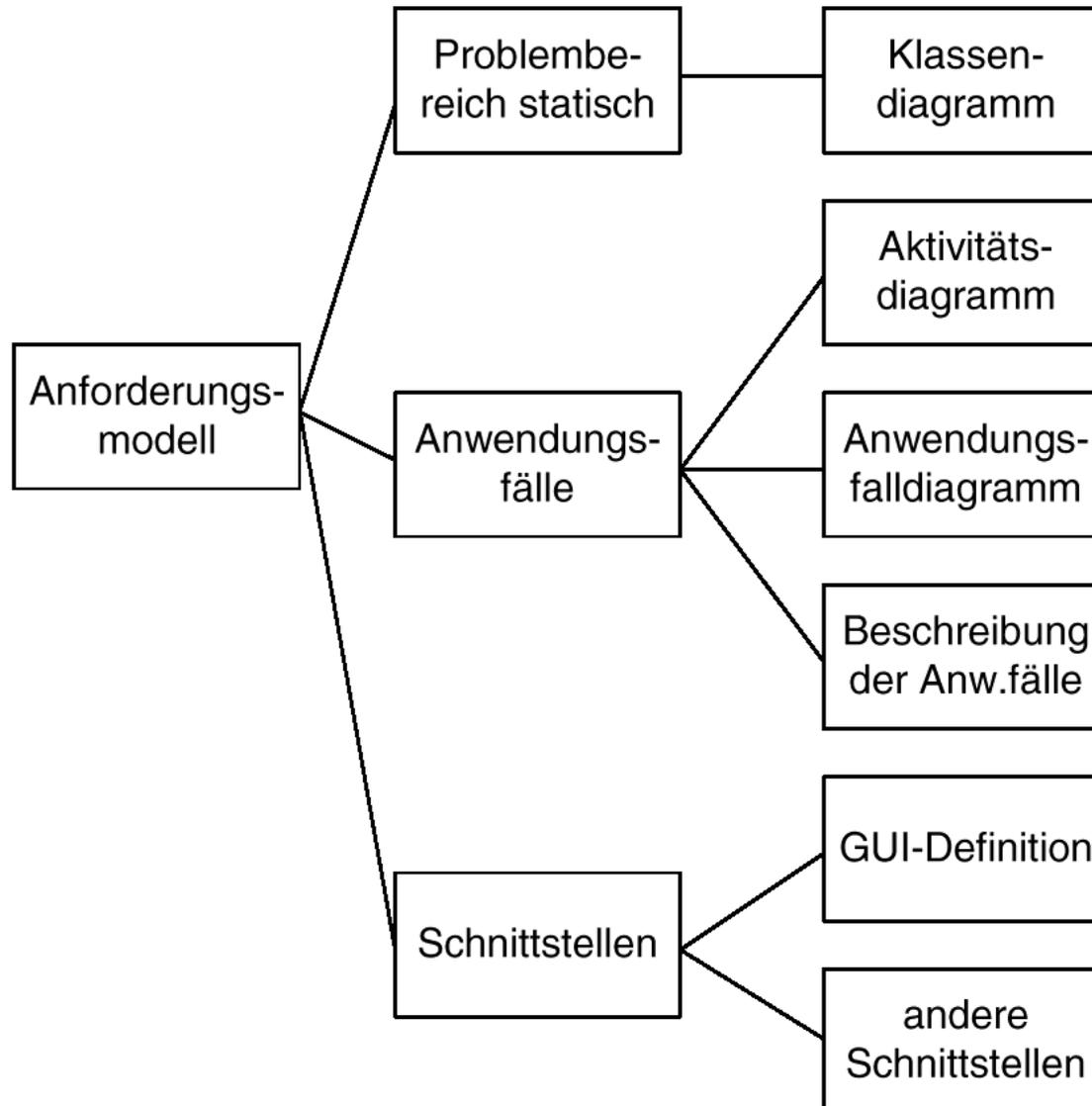
---

## Durchführung, wenn...

- Technisch realisierbar
- Erforderliches Know-How zur Realisierung beim Auftragnehmer vorhanden
- Organisatorisch beim Auftragnehmer realisierbar (Personal verfügbar)
- Kosten/Nutzen-Analyse positiv
- Rechtlich zulässig (z.B. Datenschutz beachtet)
- Keine ethisch/moralischen Bedenken durch Auftragnehmer/Entwickler



# Planungsphase: Artefakte



- Aufgabe
  - » Zusammenfassung aller fachlichen Basisanforderungen; erstes Dokument, das Anforderungen beschreibt
- Umfang
  - » wenige Seiten; gut lesbar gegliedert
- Inhalt
  - » "was", nicht "wie"; verbale und grafische Spezifikationen auf angepaßtem Abstraktionsniveau

- 1. Zielbestimmung**
  - 1.1 Musskriterien
  - 1.2 Wunschkriterien
  - 1.3 Abgrenzungskriterien  
(was nicht erforderlich ist)
  
- 2. Produkt-Einsatz**
  - 2.1 Anwendungsbereiche
  - 2.2 Zielgruppen
  - 2.3 Betriebsbedingungen

**Juristisches  
Dokument:**

Vertrag  
zwischen  
Auftraggeber  
und Auftrag-  
nehmer



## 3. Produkt-Umgebung

3.1 Software

3.2 Hardware

3.3 “Orgware“

3.4 Produkt-Schnittstellen

## 4. Produkt-Funktionen

Je Funktion ein Unterkapitel.

Funktionen aus Benutzersicht beschreiben  
(WAS geleistet wird und *nicht* WIE)





# Pflichtenheft

---

- 5. Produkt-Daten**
- 6. Produkt-Leistungen**
- 7. Benutzeroberfläche**  
Bildschirmlayout, Drucklayout,  
Tastaturbelegung, Dialogstruktur, Ton
- 8. Qualitäts-Zielbestimmung**
- 9. Globale Testszenarien**



## **10. Entwicklungs-Umgebung**

10.1 Software

10.2 Hardware

10.3 “Orgware“

10.4 Entwicklungs-Schnittstellen

## **11. Ergänzungen/Sonstiges**



## Definiert eine einheitliche Terminologie

- Beispiel:
  - » **Kundensachbearbeiter**  
Verantwortlich für die Kommunikation mit →Kunden und →Firmen einschließlich der Auskunftserteilung und Buchung
- Verwendung von branchenüblichen Begriffen, die für den Produkt-Benutzer verständlich sind
- Die Glossarbegriffe werden sowohl für die Benutzungsoberfläche als auch für die Online-Hilfe und das Benutzerhandbuch verwendet.



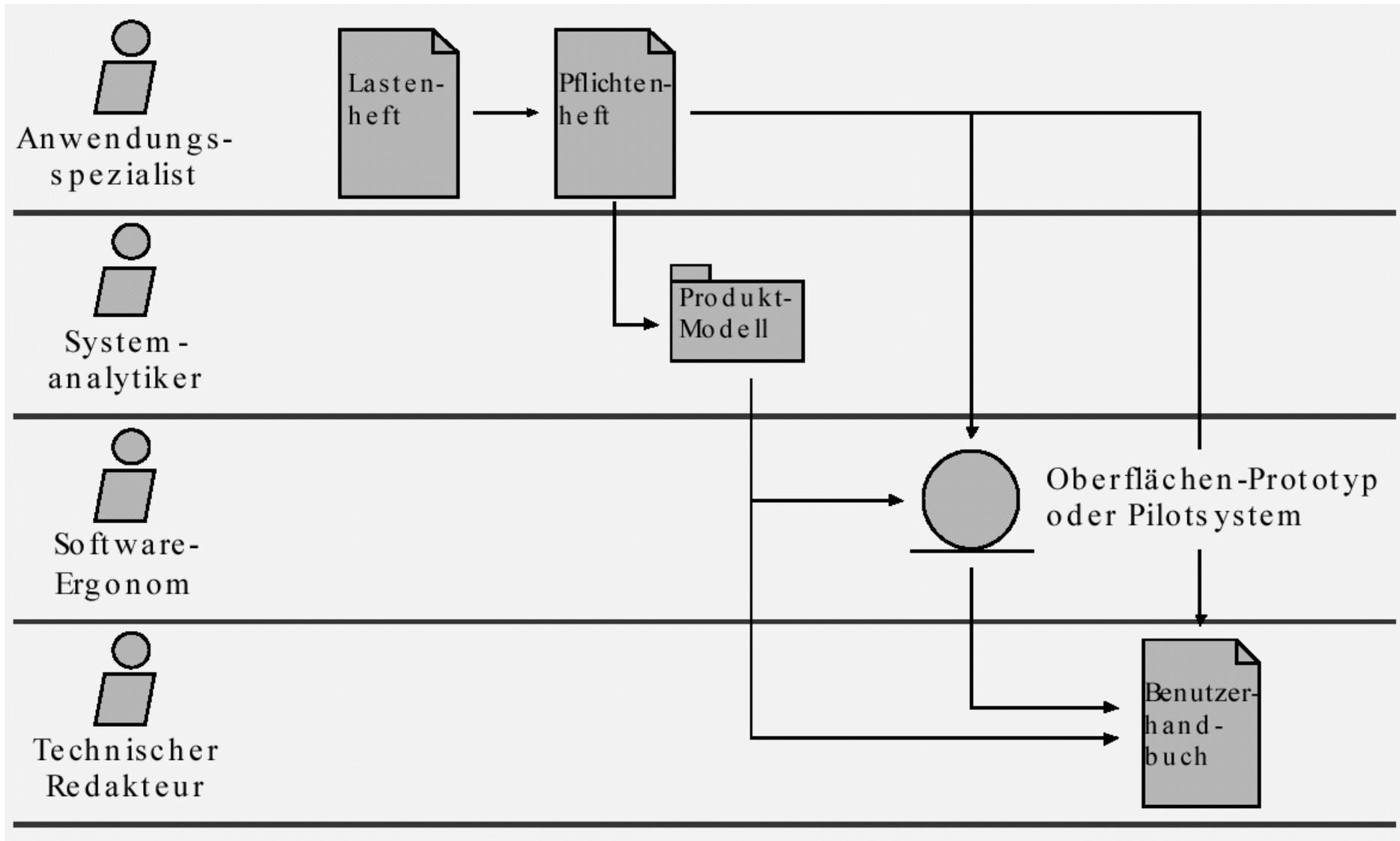
# Benutzerhandbuch

---

- Aufgabe
  - » Handhabung des Softwareprodukts beschreiben
- Adressaten
  - » Endbenutzer
  - Gibt es verschiedene Klassen von Endbenutzern (z. B. Anwender, Systemverwalter), so sollten getrennte Benutzerhandbücher geschrieben werden.
- Stile
  - » User Guide
  - » Reference Card
  - » Tutorial
  - » Online-Hilfe



# Planungsphase (Übersicht)



## Aufgabe

- Verstehen der Problemdomäne

## Relevante Artefakte

- z.B., Klassendiagramme
  - » Fachkonzepte

## Meilenstein

- Spezifikation des Problems /  
Analysemodell

## Aufgabe

- Lösungsfindung (Architektur & Details)

## Relevante Artefakte

Struktur

Dynamik

- z.B., Klassen & Interaktionsdiagramme,
  - » generelle Architektur (grob-granular)
  - » detaillierte Entscheidungen (fein-granular)

## Meilenstein

- Spezifikation der Lösung



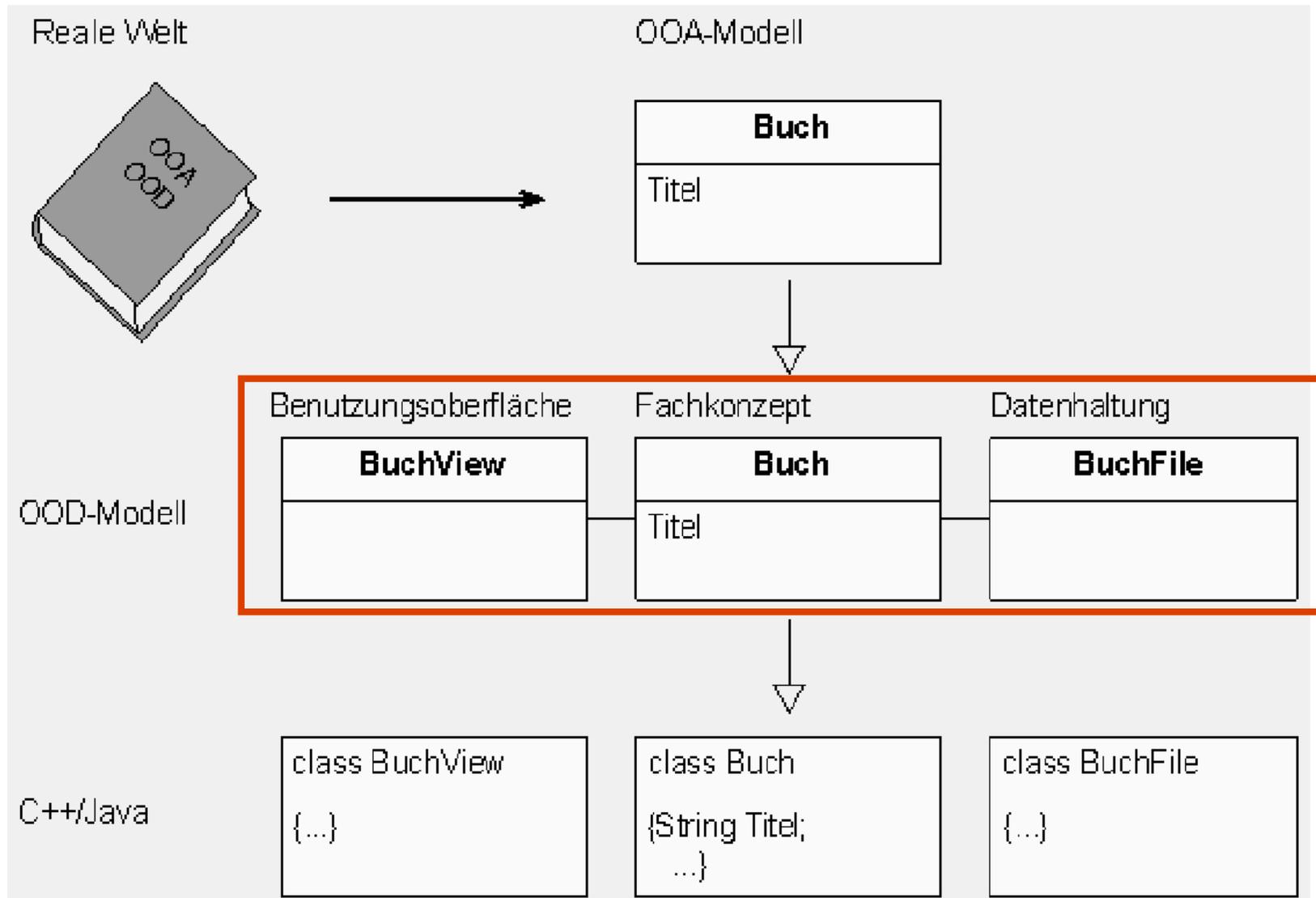
# Analyse versus Entwurf

---

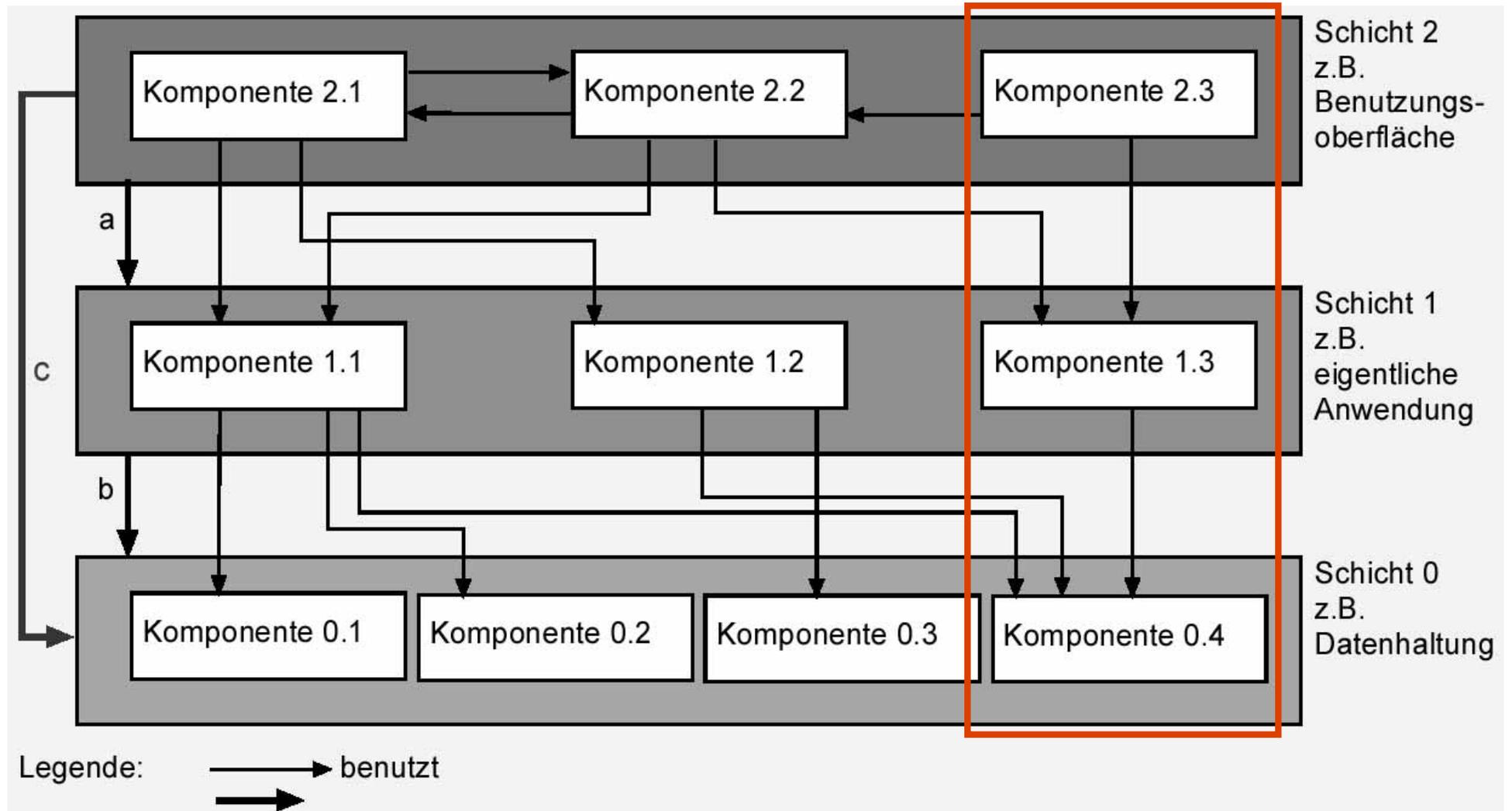
***Less than 10% of the code has to do with the purpose of the system; the rest deals with input, output, data validation, and other housekeeping.***

**Mary Shaw**

# Entwurfsaspekte



# Entwurfsaspekte





# Implementierung

---

## Aufgabe

- Umsetzung der Spezifikation in Code

## Relevante Artefakte

- z.B., Javaklassen
- Benutzerhandbuch

## Meilenstein

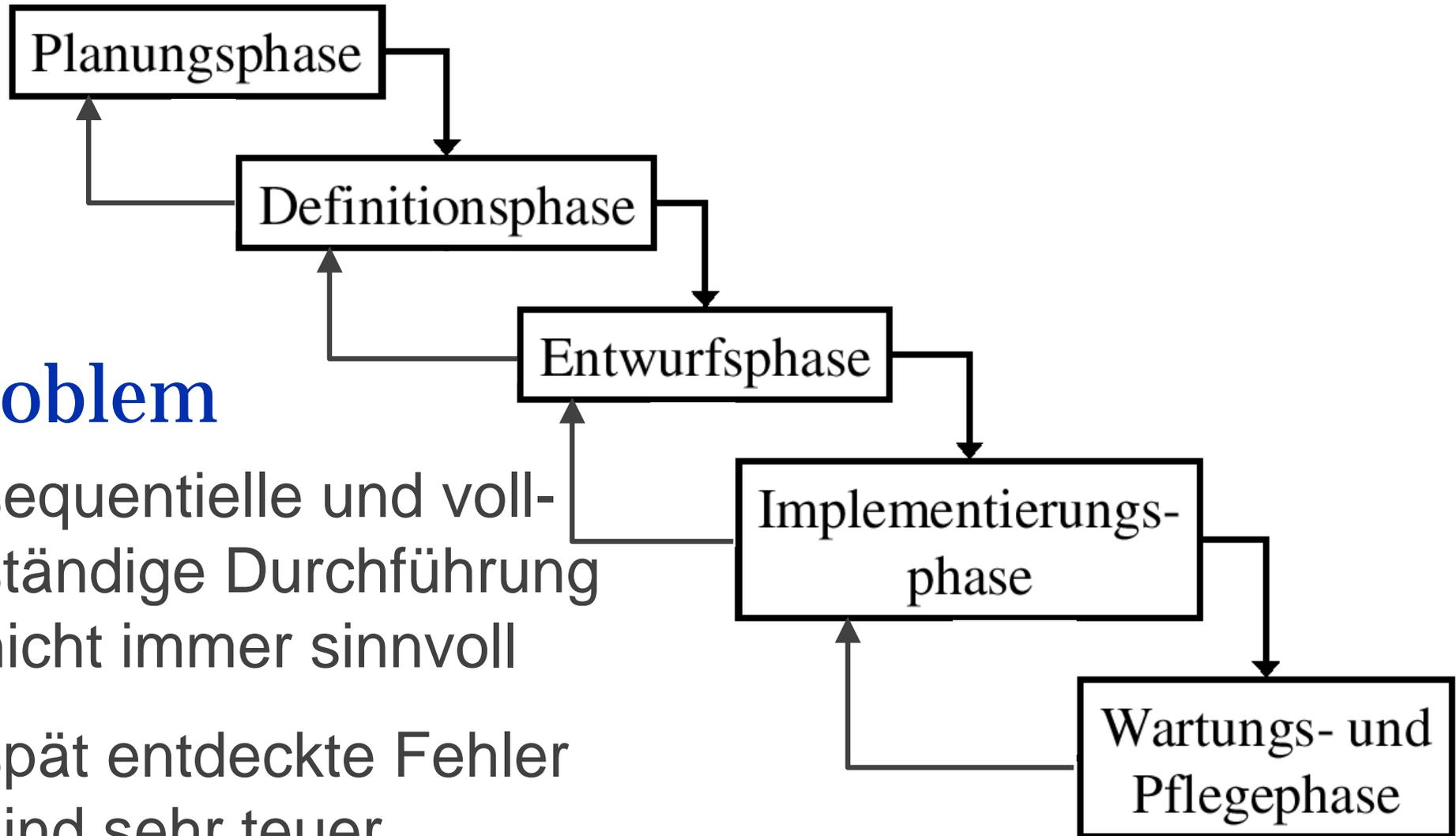
- lauffähiges System



# Phasenmodelle

---

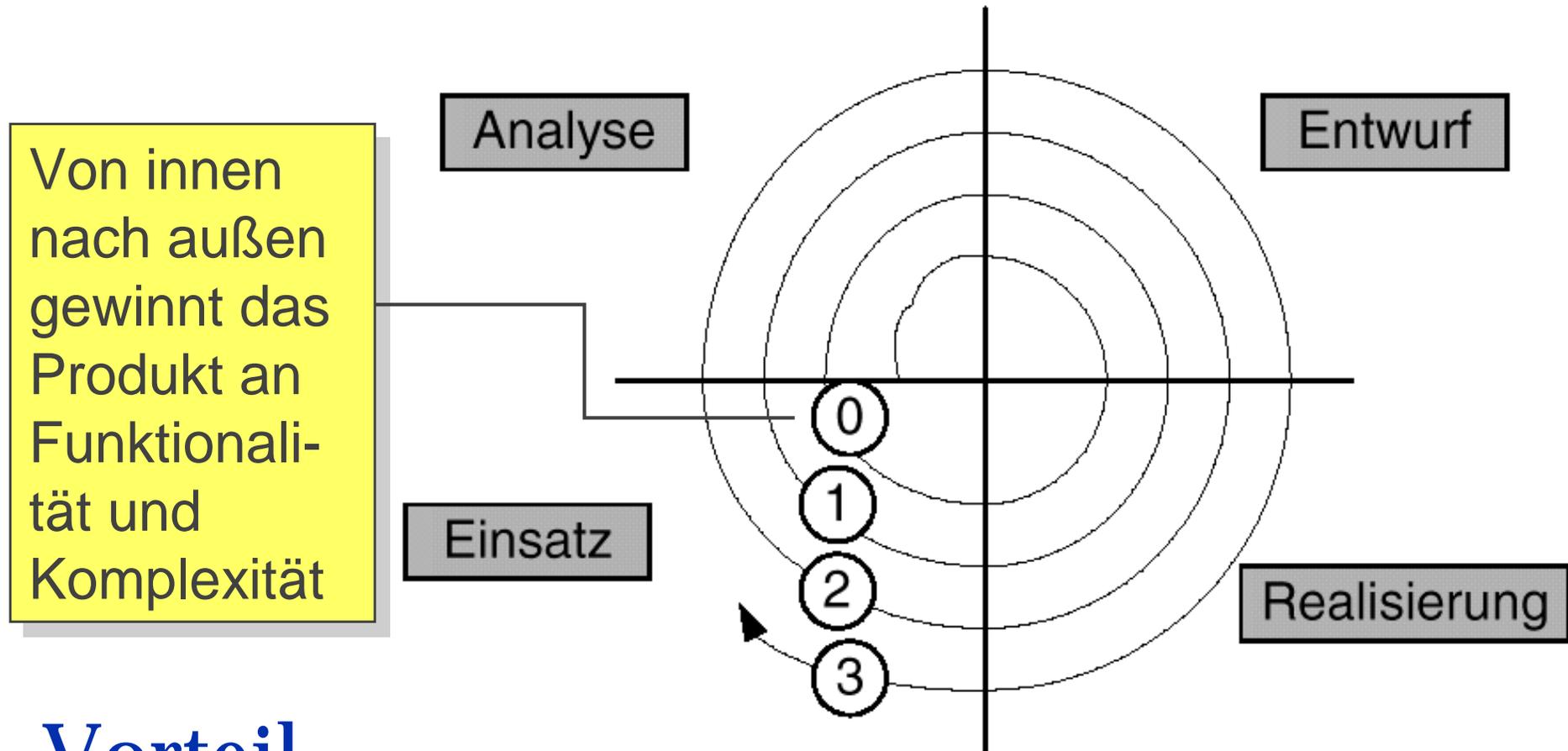
- Systematisches Vorgehen zur Entwicklung des eigentlichen Softwareprodukts
- Verschiedene Modelle kommen zum Einsatz, z.B.,
  - » Wasserfallmodell
  - » Spiralmodell
  - » Versionsmodell
  - » Iteratives Phasenmodell
  - » Prototypenmodell



## Problem

- sequentielle und vollständige Durchführung nicht immer sinnvoll
- spät entdeckte Fehler sind sehr teuer

# Spiralmodell



## Vorteil

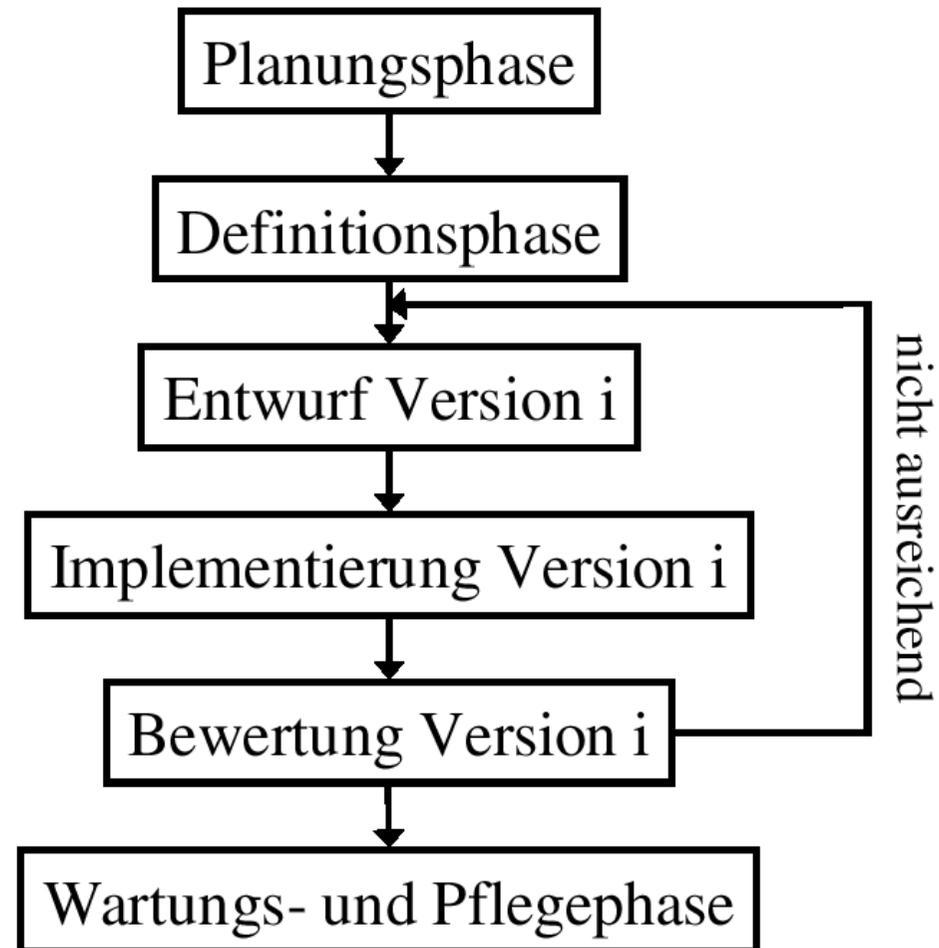
- Frühe Rückmeldung aus späteren Phasen

## Vorteil

- Auftraggeber erhält jeweils einsatzfähige Produkte

## Nachteil

- Gefahr der Komplett-überarbeitung

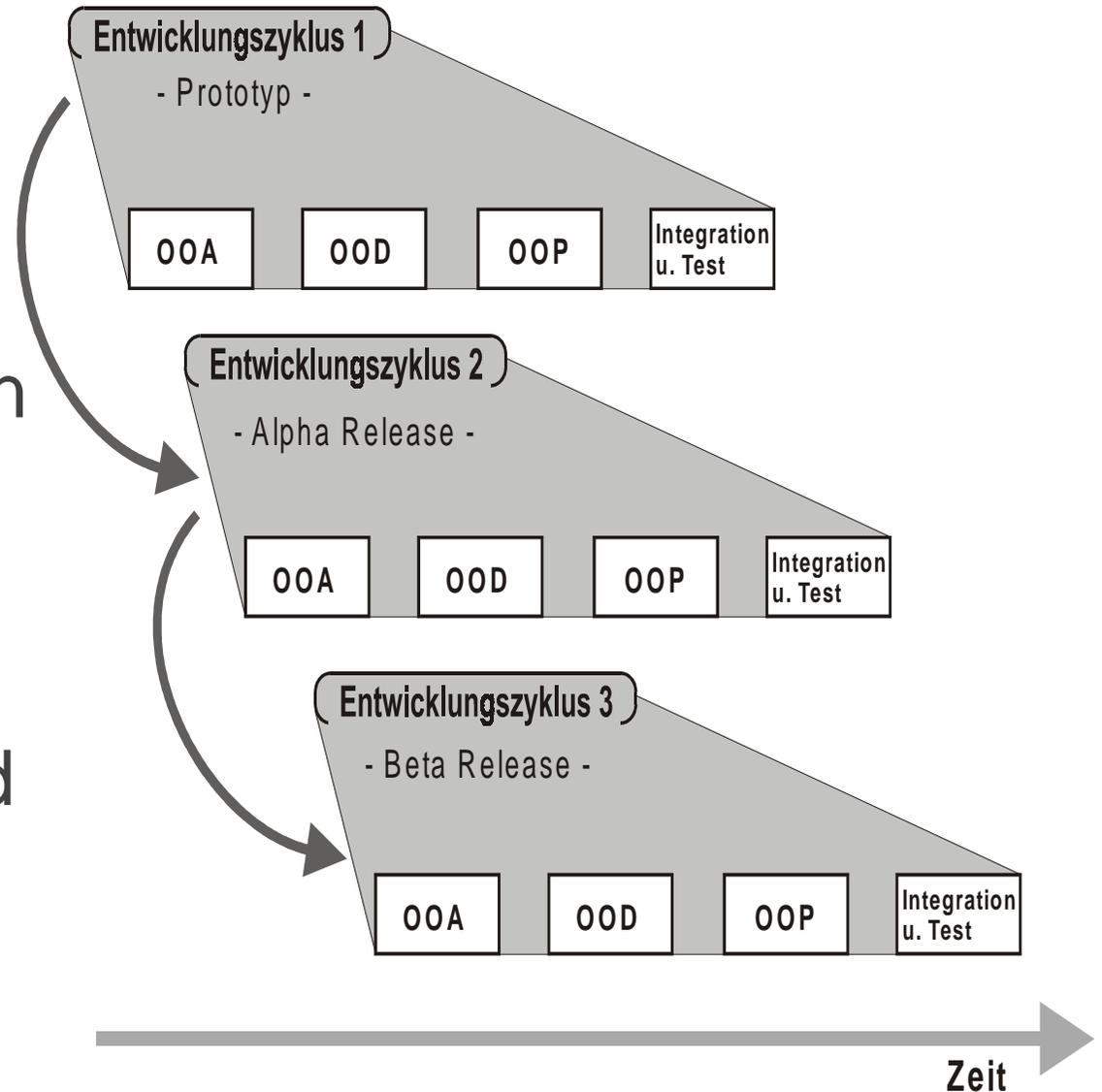


## Vorteil

- Reduktion von Wartezeiten zwischen abhängigen Aktivitäten

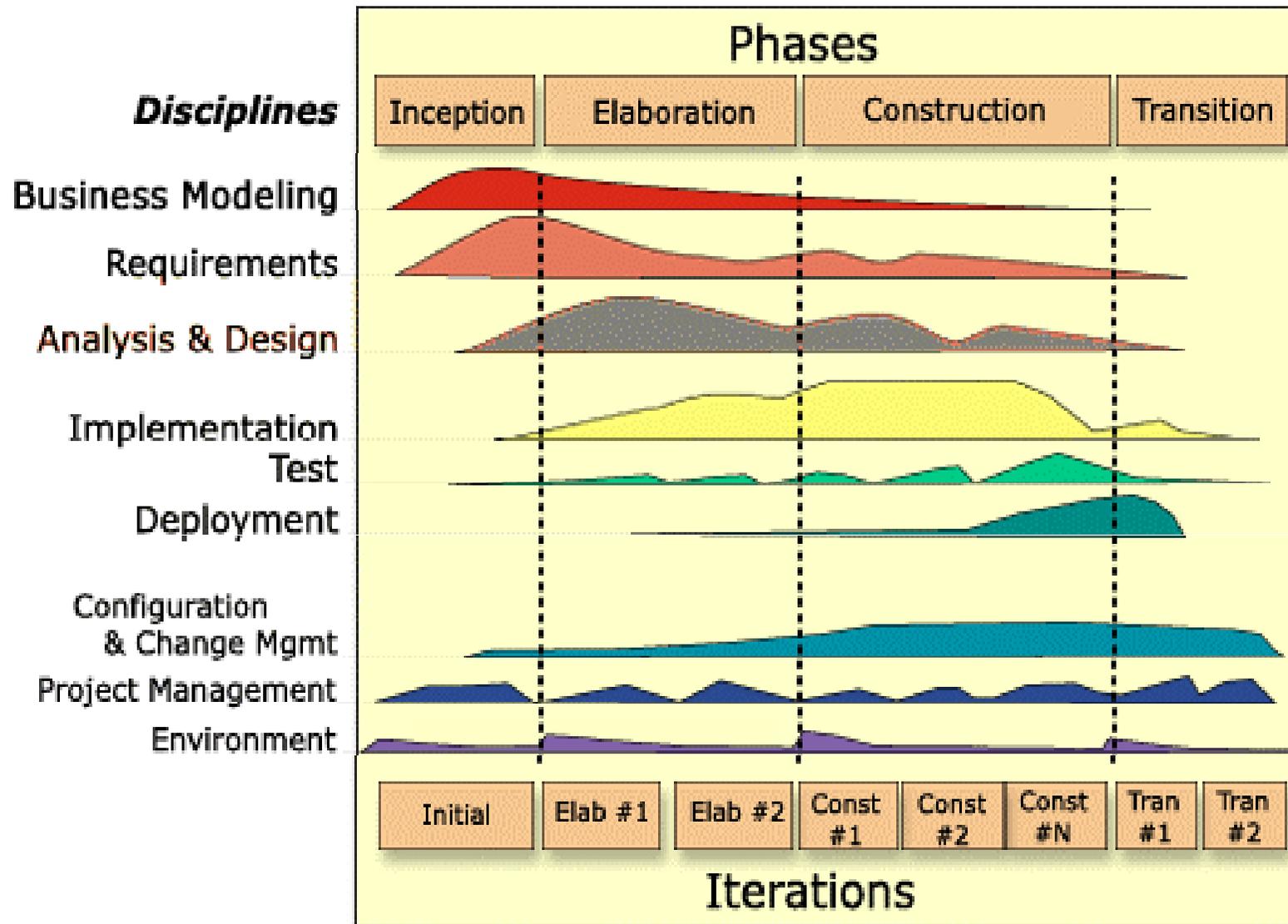
## Nachteil

- Hoher Planungs- und Personalaufwand

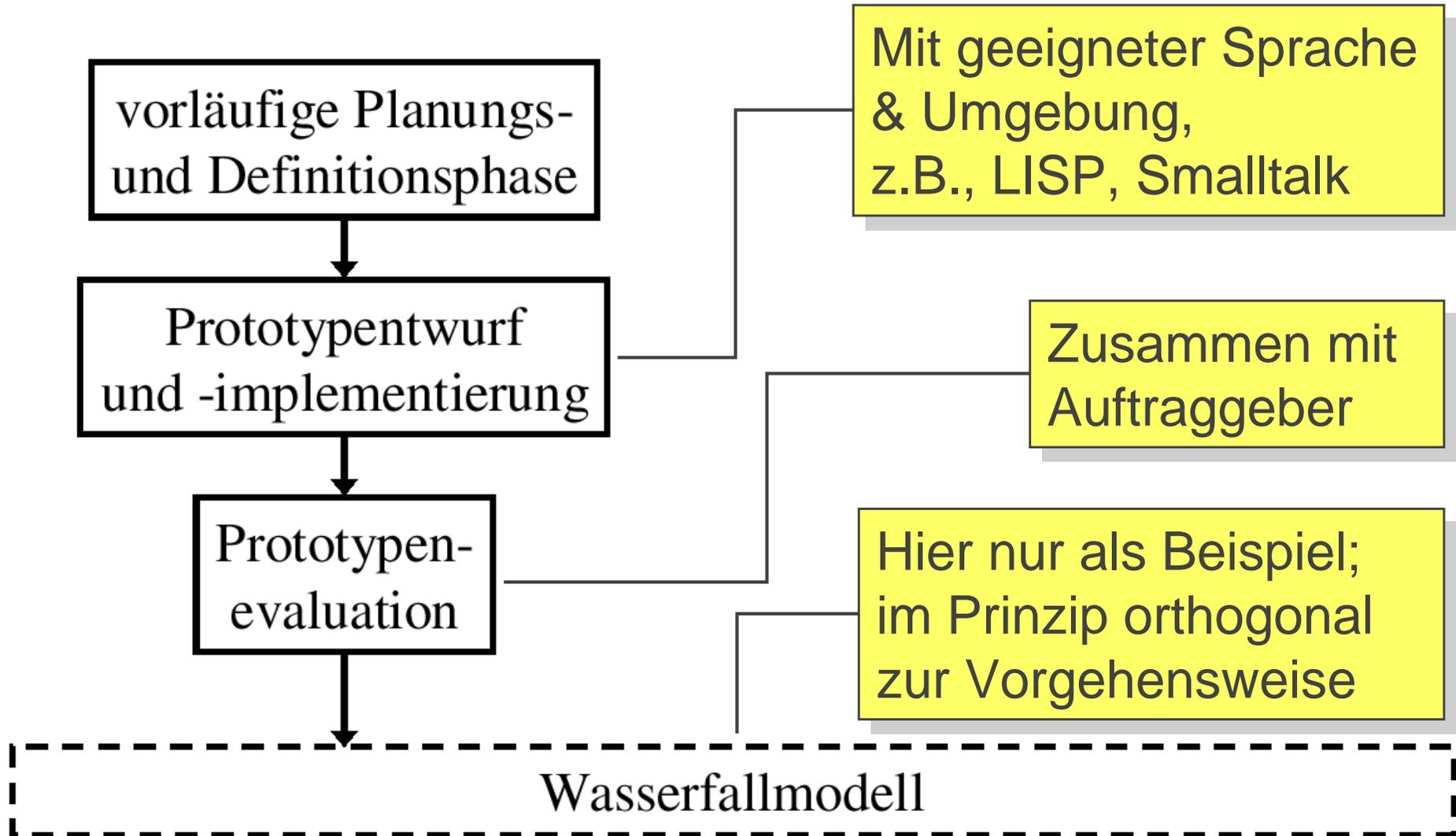




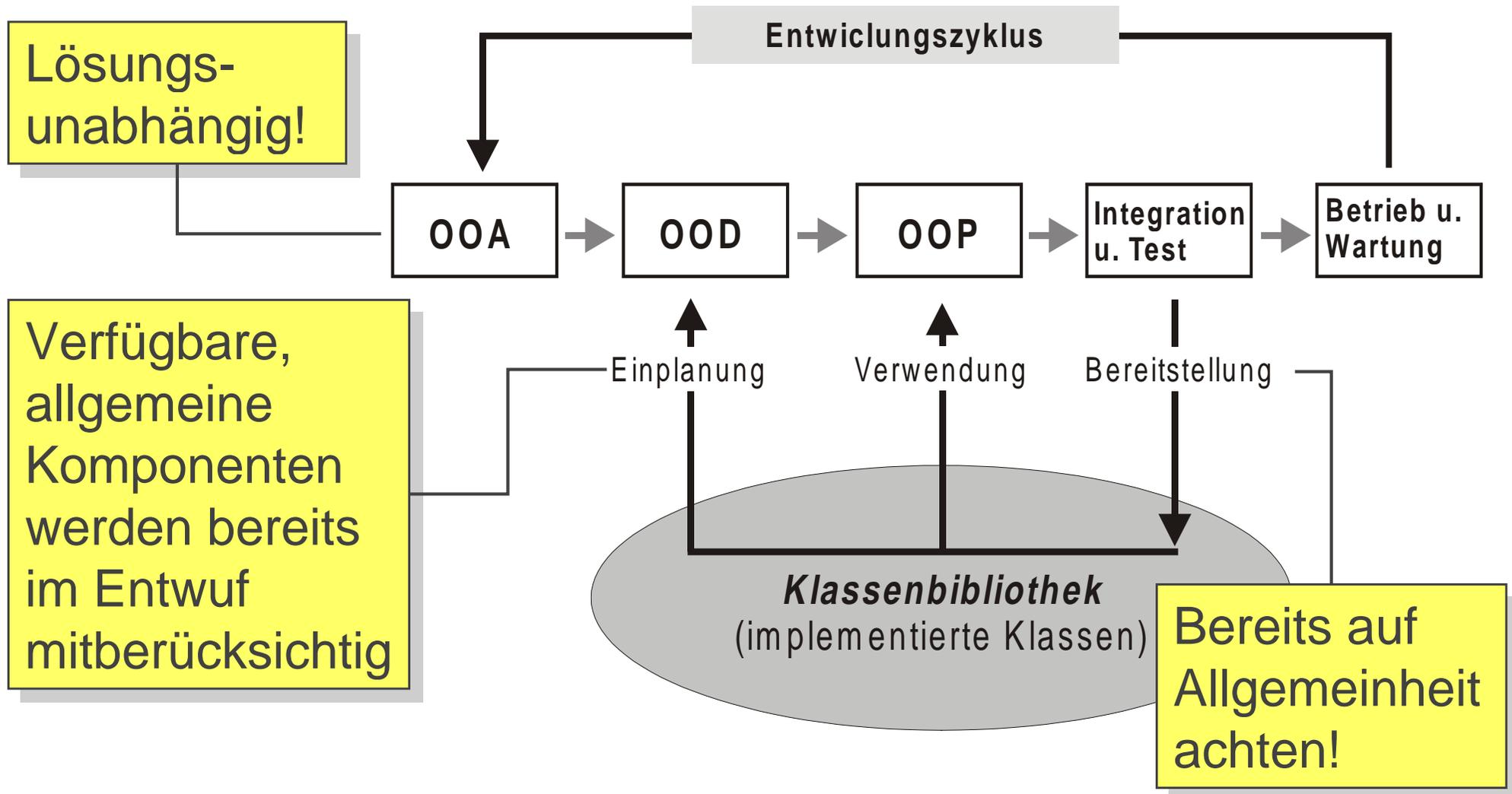
# Rational Unified Process



# Prototypenmodell



# Wiederverwendung





# "Planning for Change"

***In preparing for battle  
I have always found  
that plans are useless,  
but planning is indispensable.***

**General Eisenhower**