



# Software-Engineering / Software Qualität

## **Softwareprozesse - Teil 1**

Prof. Dr. Cathrin Möller

Prof. Dr. Martine Herpers

Jochen Hosenfeld



- **Kennen** der wichtigsten Softwareprozessmodelle
- **Verständnis** der einzelnen **Aktivitäten** und ihrer **Bedeutung**
- Kennen der **Bedeutung** von **Änderungsprozessen** in der Softwareentwicklung



- **Vorgehensmodelle**
- Prozessaktivitäten
- Umgang mit Änderungen
- Rational Unified Process



- **Vorgehensmodell/Prozessparadigma**

# Vorgehensmodelle des SE



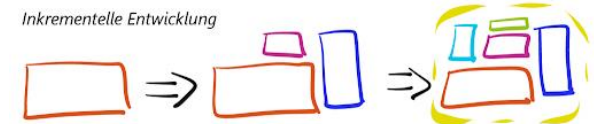
- Wasserfallmodell



- V-Modell (für IT-Projekte der öffentlichen Hand in Deutschland)



- Inkrementelle Entwicklung



- Wiederverwendungsorientiertes Software-Engineering

# Grundlegende Prozessaktivitäten



**Analyse und  
Definition der  
Anforderungen**

Funktionalität, Einschränkungen, Ziele  
zusammen mit Systembenutzern festlegen =>  
Ergebnis: Systemspezifikation

# Grundlegende Prozessaktivitäten



Analyse und  
Definition der  
Anforderungen

System- und  
Softwareentwurf

Anforderungen werden Software oder Hardware zugeordnet, Software Struktur festgelegt => Ergebnis System- und Software-Architektur

# Grundlegende Prozessaktivitäten



Analyse und  
Definition der  
Anforderungen

System- und  
Softwareentwurf

Implementierung  
und Modultests

Implementierung der Module, Tests  
der einzelnen Module => Ergebnis:  
Code-Module und Testprotokolle



# Grundlegende Prozessaktivitäten



Analyse und  
Definition der  
Anforderungen

System- und  
Softwareentwurf

Implementierung  
und Modultests

Integration und  
Systemtest

Teile werden integriert und als  
Ganzes getestet => Ergebnis: fertiges  
System/Software

# Grundlegende Prozessaktivitäten



Analyse und  
Definition der  
Anforderungen

System- und  
Softwareentwurf

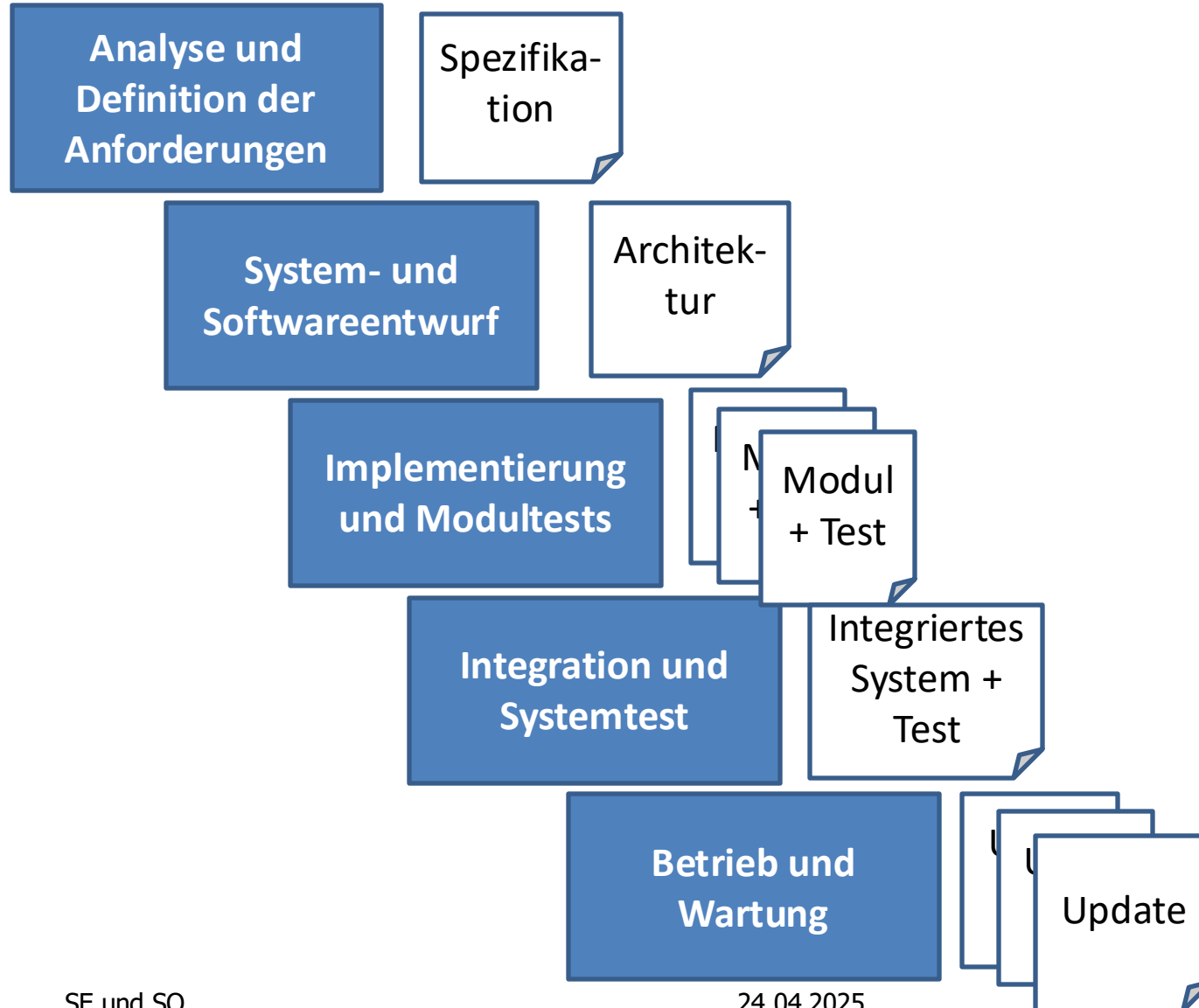
Implementierung  
und Modultests

Integration und  
Systemtest

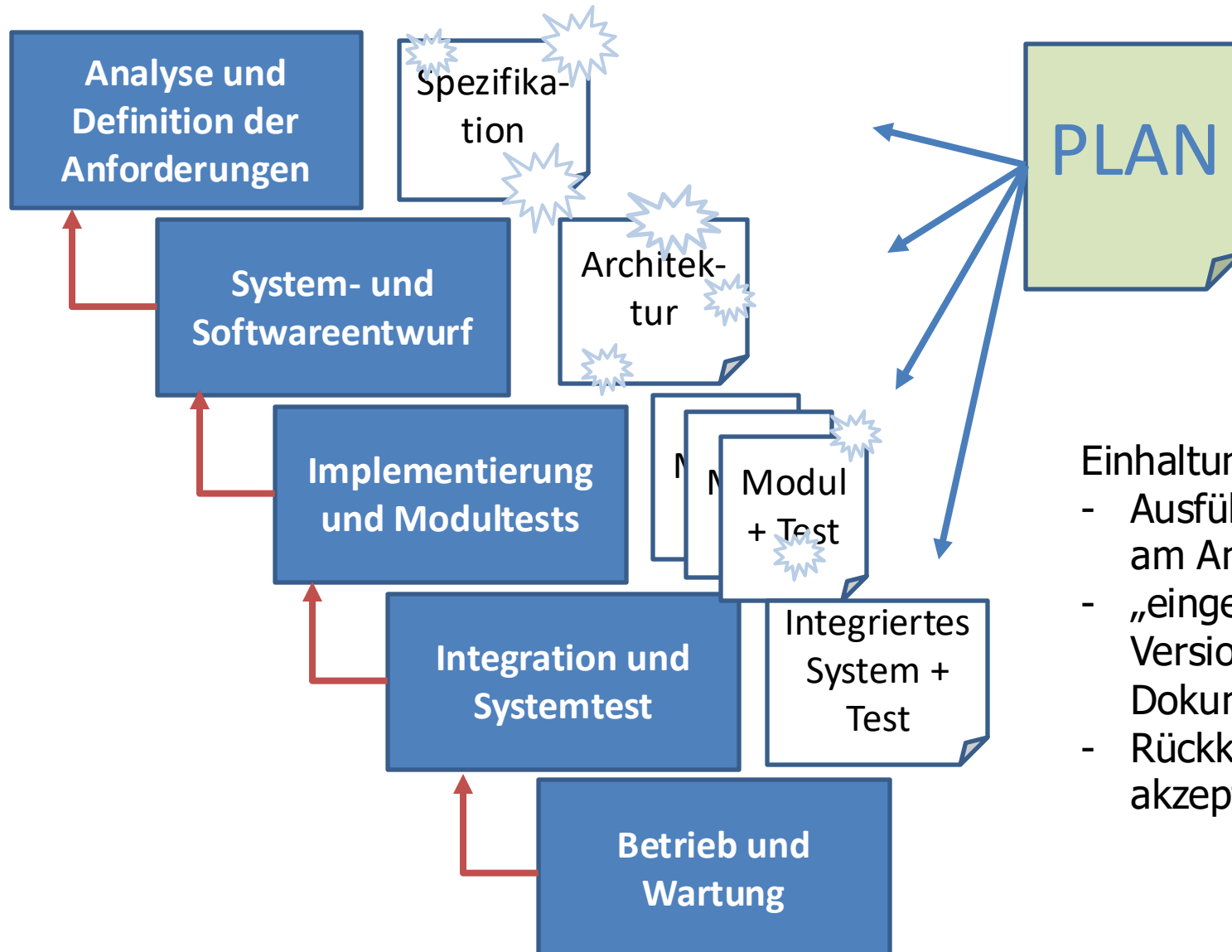
Betrieb und  
Wartung

Installation, zum Gebrauch  
freigegeben. Korrektur von  
Fehlern und Verbesserungen  
bei neuen Anforderungen.

# Dokumentensicht



# Plangesteuertes Modell



Einhaltung des Plans:

- Ausführliche Planung am Anfang
- „eingefrorene“ Versionen der Dokumente (offiziell)
- Rückkopplungen akzeptieren



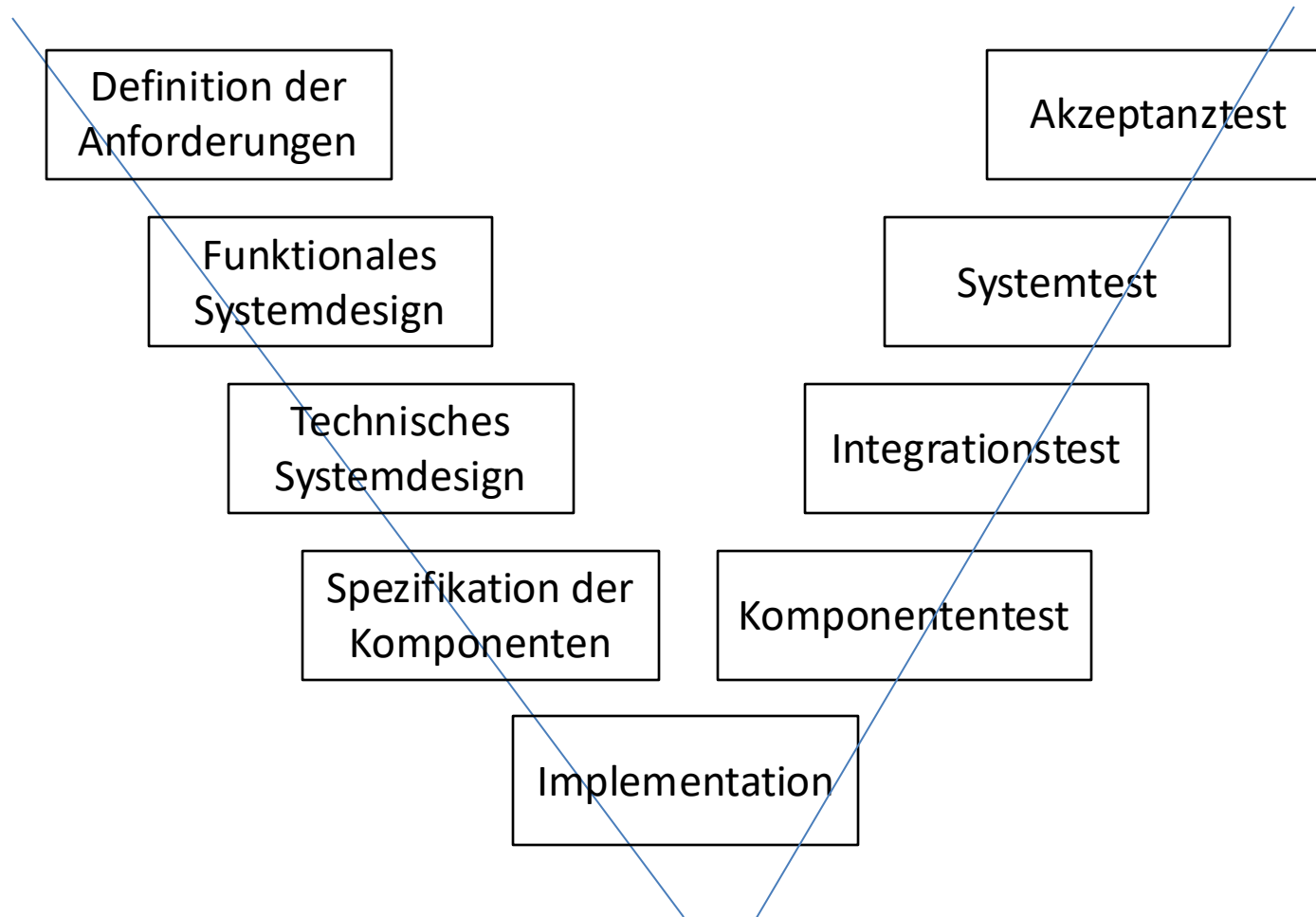
# Wasserfallmodell-Varianten

- Formale Systementwicklung für sicherheitskritische System
  - B-Methode (Pariser Metro, Linie 14)
  - Clean-Room-Prozess (IBM – Ziel: Null-Fehler-Software)



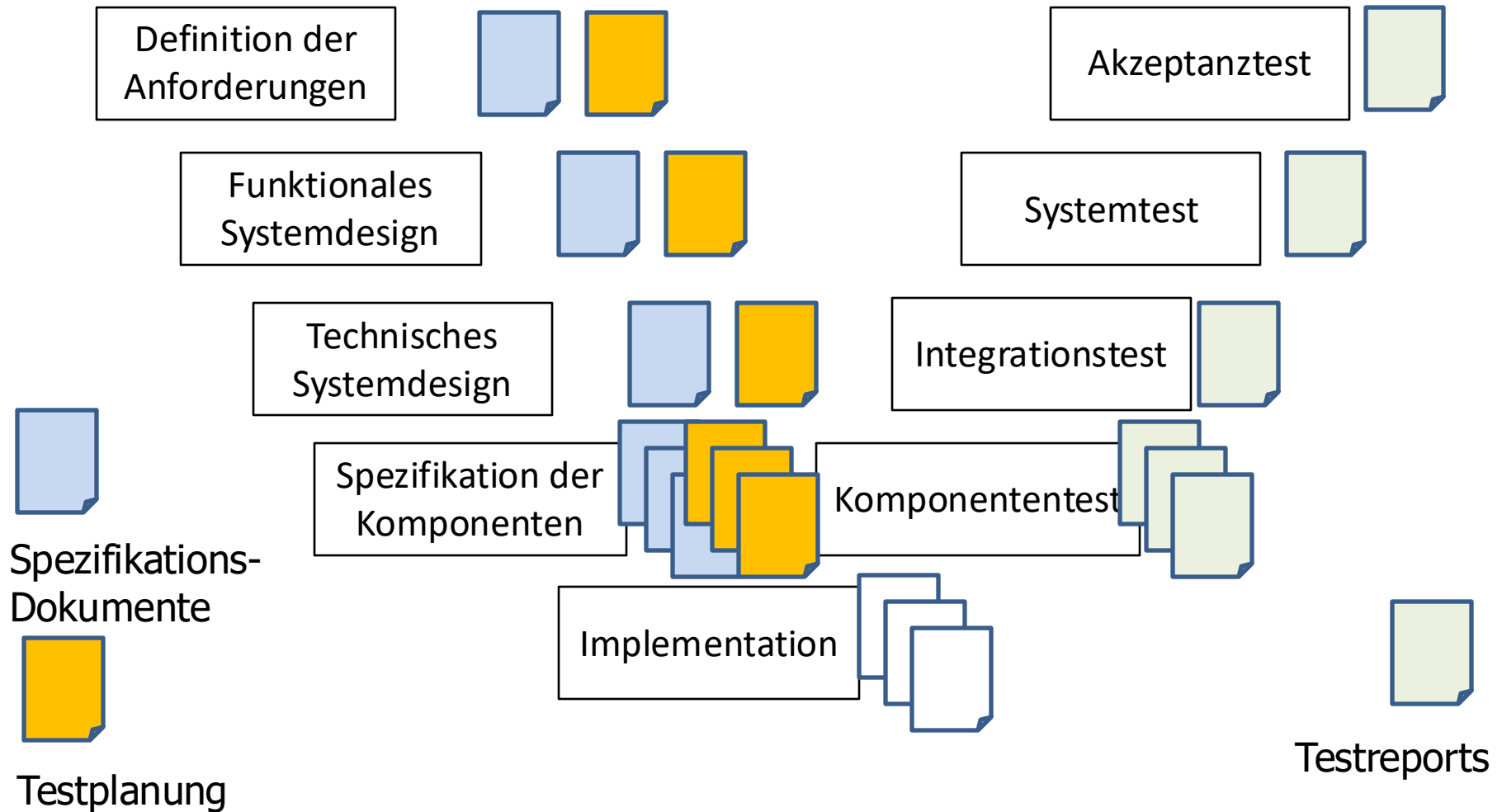
# V-Modell - Phasenmodell

Ähnlich dem Wasserfallmodell



Erklärungsvideo: <https://www.youtube.com/watch?v=5ahIznIqFyo>

# V-Modell - Dokumentensicht





# V-Modell vs. Wasserfallmodell

- Treffen die Aussagen unten auch für das V-Modell zu?
  - Plangesteuert
  - dokumentenlastig: Erstellung und Abnahme
  - Wenige Rückkopplungen erlaubt
  - Eingefrorene Dokumente für die Planeinhaltung





# V-Modell vs. Wasserfallmodell

- Treffen die Aussagen unten auch für das V-Modell zu?
  - Plangesteuert
  - dokumentenlastig: Erstellung und Abnahme
  - Wenige Rückkopplungen erlaubt
  - Eingefrorene Dokumente für die Planeinhaltung

**Ja, kein Unterschied  
zum Wasserfallmodell**



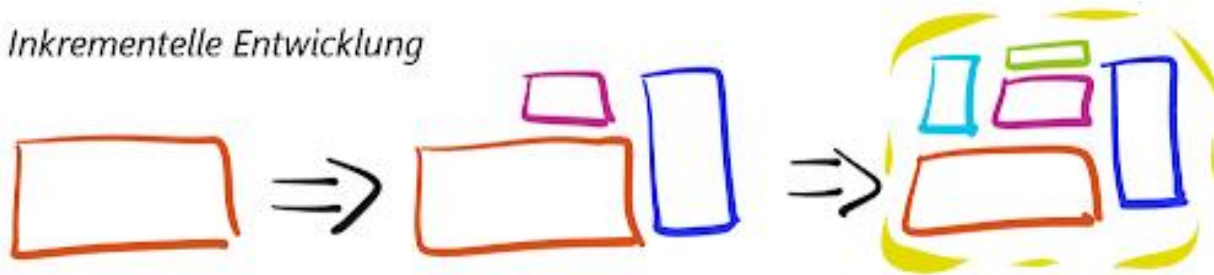
# Unterschiede V- vs. Wasserfall

- Beides sind Phasenmodelle:
  - einzelne in sich abgeschlossene Stufen
  - Einmaliger Durchlauf
  - Plangesteuert
  - dokumentenlastig
- Zusätzlich im V-Modell
  - Augenmerk auf das Testen
    - Zwei Zweige: Konstruktion und Test
    - Frühzeitige Planung der Tests



# Inkrementelle Entwicklung

*Inkrementelle Entwicklung*

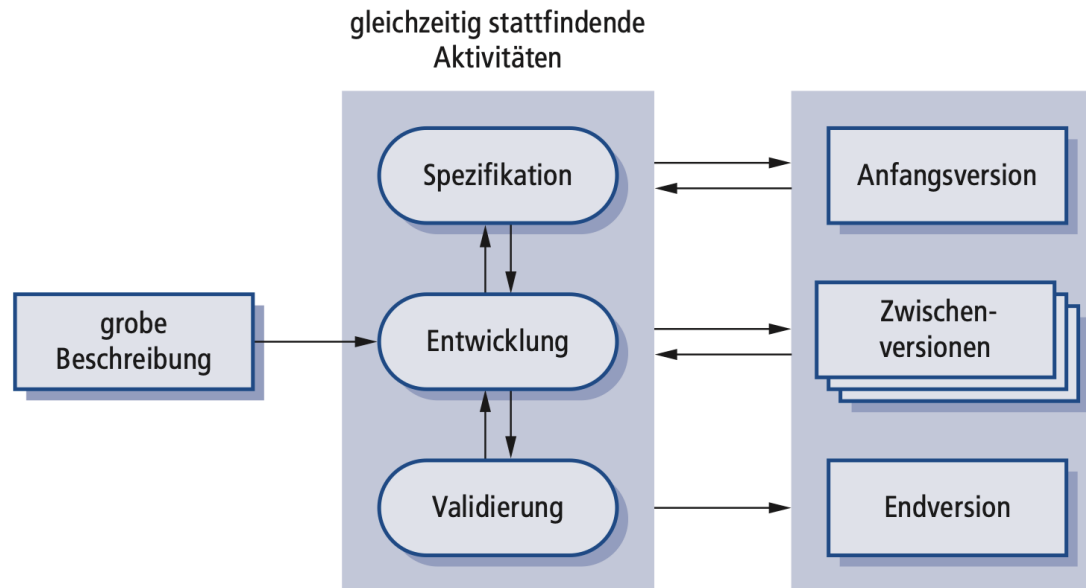


- Erste Version
- Kommentare von Kunden/Kundinnen
- Schrittweise Weiterentwicklung

z.B.:



# Inkrementelle Entwicklung



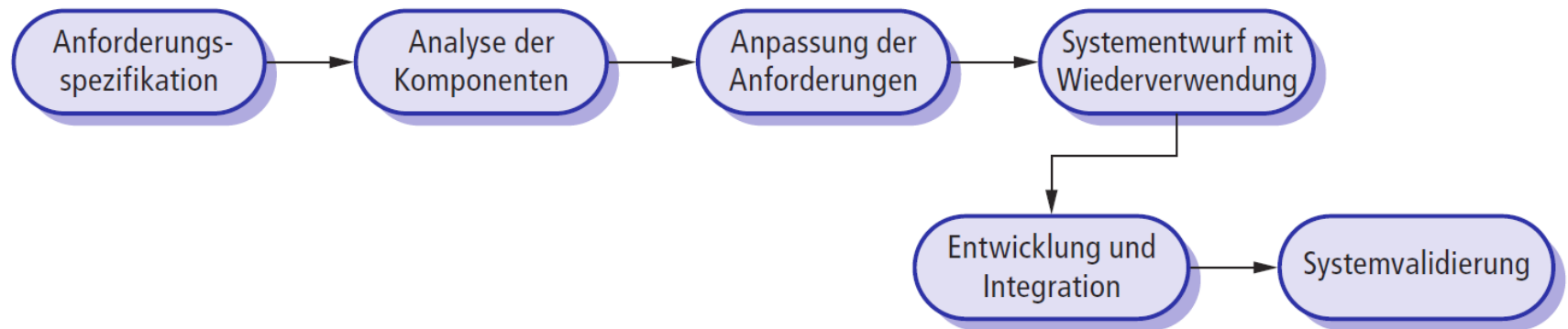
Quelle: Ian Sommerville, Software Engineering, PEARSON, 2012



# Inkrementelle Entwicklung

- Überlegen Sie mögliche Vor- und Nachteile gegenüber den plangesteuerten Phasenmodellen.
- Kriterien für die Bewertung der Modelle:
  - Zuverlässigkeit der Planung
  - Stabilität der Kundenwünsche
    - Unwissenheit über technische Möglichkeiten
    - sich ändernde Kundenwünsche (z.B. durch neue Technologien)
  - Schnelle Nutzbarkeit für Kunden/Kundinnen
  - Kosten für die Dokumentation
  - Qualität des Produkts

# Wiederverwendungsorientiertes Software-Engineering



Aktivitäten-Diagramm

Quelle: Ian Sommerville, Software Engineering, PEARSON, 2012

# Wiederverwendungsorientiertes Software-Engineering



- Vorteile:
  - Reduktion von Kosten für Neuentwicklung
  - Reduktion der Risiken
- Nachteile:
  - Eventuell können nicht alle Anforderungen umgesetzt werden

# Wiederverwendung



Bsp: Komponenten / Software muss erweitert werden um optionale Felder

## Changes for reusability

HOCHSCHULE FULDA  
UNIVERSITY OF APPLIED SCIENCES

Extending data fields for addresses => optional / identifier for country

```
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.Id;
import jakarta.validation.constraints.NotBlank;
```

```
@Entity
public class Address {

    @Id
    @GeneratedValue
    private int id;

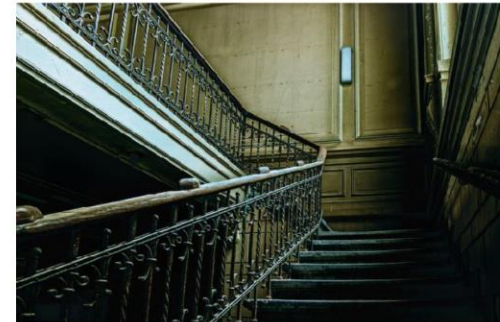
    @NotBlank
    private String street;
    @NotBlank
    private String houseNumber;
    @NotBlank
    private String postalCode;
    @NotBlank
    private String city;
```



```
@Entity
public class Address {

    @Id
    @GeneratedValue
    private int id;

    @NotBlank
    private String street;
    @NotBlank
    private String houseNumber;
    @NotBlank
    private String stiege;
    @NotBlank
    private String postalCode;
    @NotBlank
    private String city;
    @NotBlank
    private String country;
```



Prof. Dr. Cathrin Möller - Angewandte Informatik



# Wiederverwendung



Bsp: Konfigurierbarkeit

## Industry example - CBSE with reuse

HOCHSCHULE FULDA  
UNIVERSITY OF APPLIED SCIENCES 

### Car vs Bike Insurance Portal / Health vs Pet Insurance Portal

- base data handling of **user, address, contract** identical
- **payment** handling identical
- different "**product**" pages describing the specific insurance
- different **fields / attributes** as options for the contract to be selected



Prof. Dr. Cathrin Möller - Angewandte Informatik

# Wiederverwendung

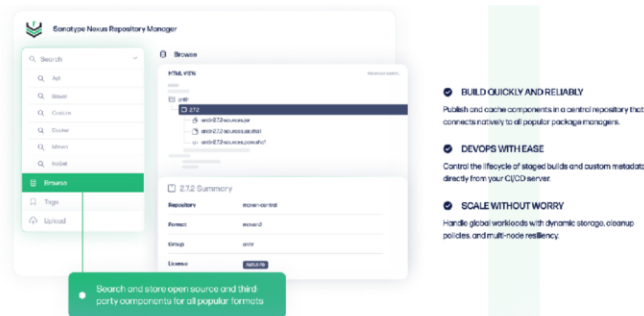


"Pool" an vorhandener (Firmen-privater) Software vs public / open source

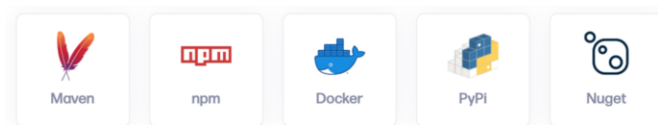
## Reuse from company Nexus

HOCHSCHULE FULDA  
UNIVERSITY OF APPLIED SCIENCES

<https://www.sonatype.com/products/sonatype-nexus-repository>



Big companies usually host a Nexus repository where all packages can be shared within the same without publishing them to the public NPM / Maven / ... repo



Although e.g. NPM has "company private" package hosting solutions as well



Prof. Dr. Cathrin Möller - Angewandte Informatik



- Vorgehensmodelle
- **Prozessaktivitäten**
- Umgang mit Änderungen
- Rational Unified Process