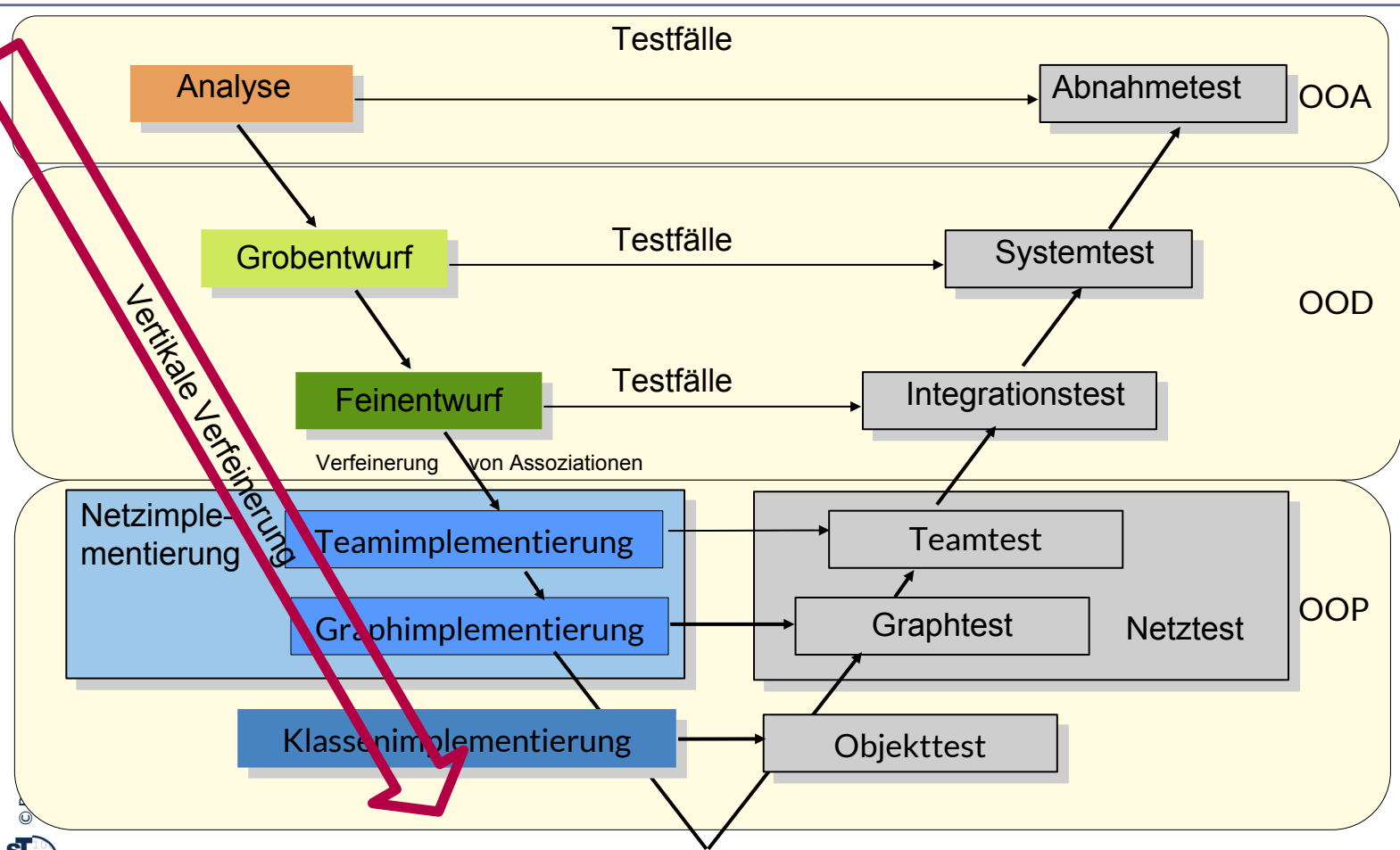


5.5

# Q4: Softwareentwicklung im V-Modell

[Boehm 1979]



## Modellierung im V-Modell

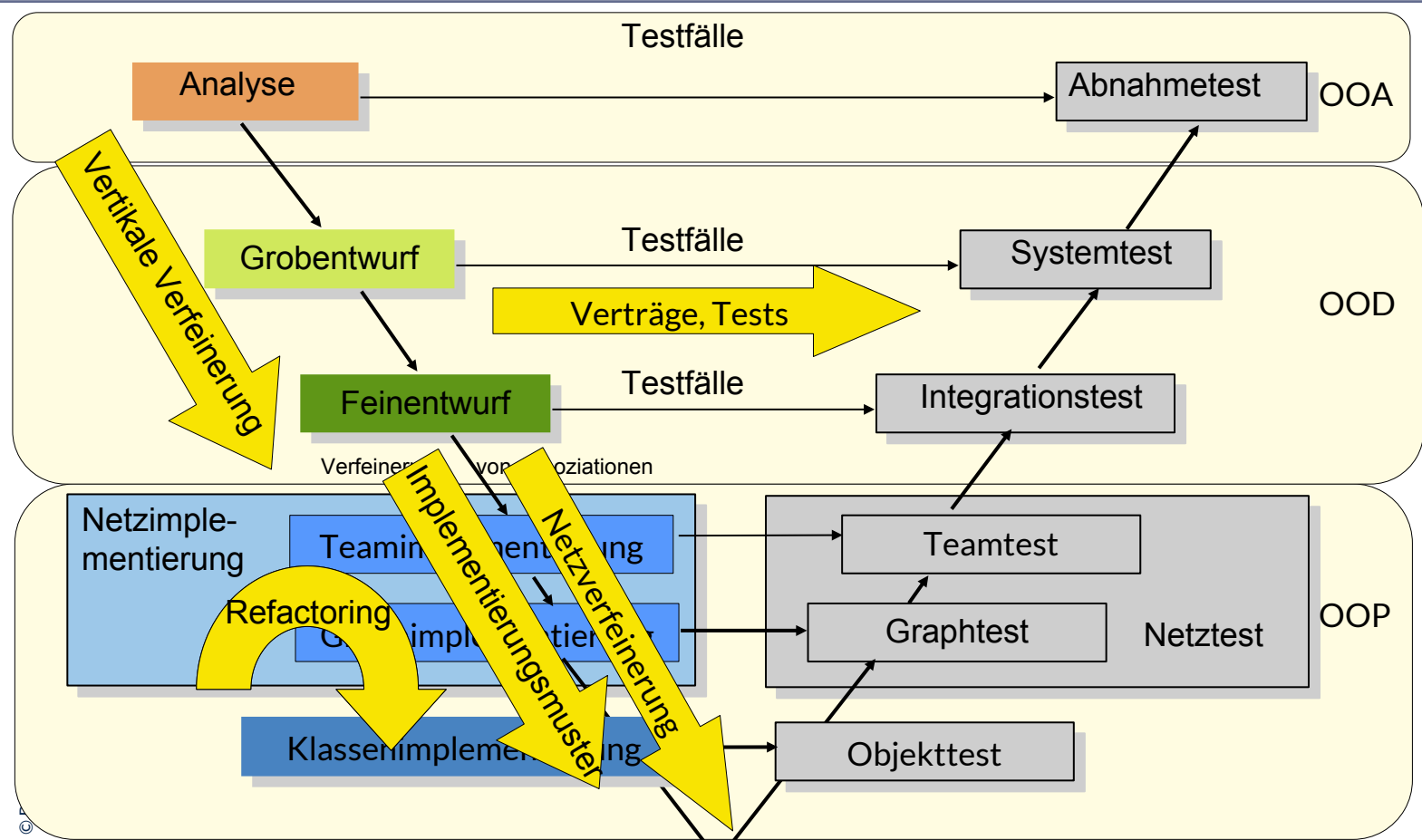
- ▶ Im V-Modell werden mehrere Sprachen gleichzeitig benutzt (hier aUML, dUML, jUML)
  - ▶ Die Modelle werden durch verschiedene Verfeinerungsoperationen aus dem initialen Anforderungsmodell entwickelt
- ▶ Ein **Sprachkonstrukt (Sprachelement)** bezeichnet ein Konstrukt bzw. Konzept einer Sprache.
  - ▶ Ein **Programm-/Modellelement** bezeichnet ein Element eines Programms bzw. eines Anforderungs- oder Entwurfsmodells.
  - ▶ Ein **Fragment (Snippet)** eines Programms oder Modells ist ein partieller Satz der Sprache, d.h. ein Netz aus Programm- oder Modellelementen.
  - ▶ Ein **generisches Fragment (generisches Snippet, Fragmentformular)** eines Programms oder Modells ist ein partieller Satz der Sprache mit Platzhaltern ("Lücken").
  - ▶ Eine **Fragmentgruppe** ist eine Menge von (ggf.generischen) Fragmenten eines Programms oder Modells.
  - ▶ Eine **Fragmentkomponente** ist eine Fragmentgruppe zur Wiederverwendung.
  - **Abstraktion** ist das Vernachlässigen von Details
  - **Detailierung (Anreicherung)** ist das Anfügen von Details

# Software-Entwicklung im V-Modell

- ▶ **Horizontale Verfeinerungsoperationen** ersetzen Fragmente und Fragmentgruppen auf gleicher Sprachebene:
  - **Detaillierung (Anreicherung):** Ergänzung von Einzelheiten
    - **Vervollständigung (Elaboration)** von Fragmenten zu Sätzen der Modellierungssprache
  - **Erhöhung Zuverlässigkeit:** Ergänzung von qualitätssteigernden Fragmenten (Typisierung, Verträge, Tests)
  - Einführung des **Architektur-Aspektes** des Systems
    - **Strukturierung** und **Restrukturierung**
    - **Refaktorisierung (Refactoring)** ist semantische Restrukturierung
- ▶ **Vertikale Verfeinerungsoperationen** (von abstrakter Ebene zu konkreter Ebene) vereinfachen Fragmente und wechseln dabei oft die Sprache, z.B. von UML nach Java:
  - **Abflachen** von Fragmenten (Flachklopfen, Realisierung, lowering): Realisierung ersetzt ausdrucksstarke Konstrukte durch weniger ausdrucksstarke, implementierungsnähere
  - Einsatz von Implementierungsmustern

# Q4: Softwareentwicklung im V-Modell

[Boehm 1979]

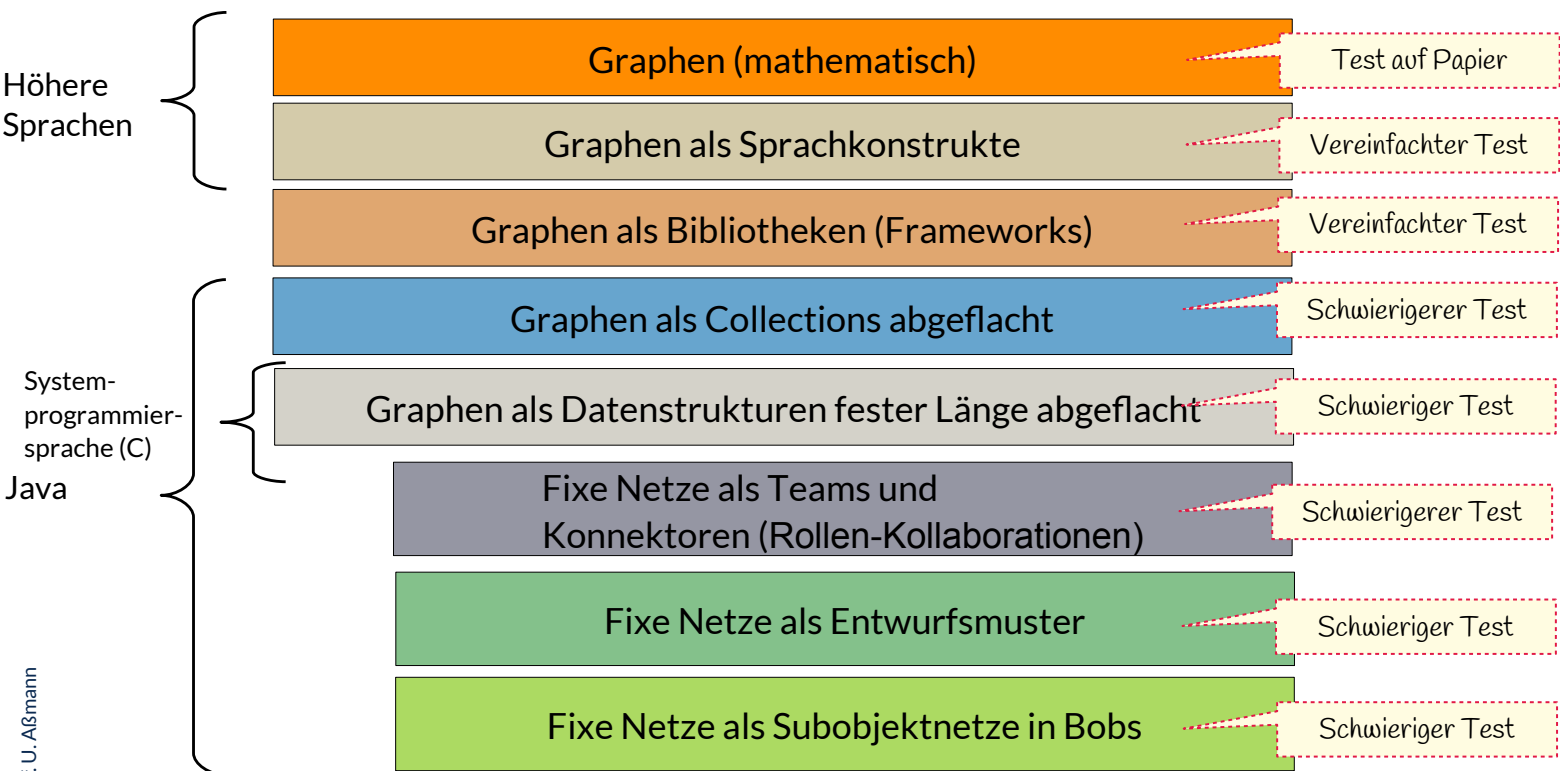


## Verfeinerung: Schritte von UML zur Implementierung

Ein **Implementierungsmuster** (*workaround, Idiom*) beschreibt die vertikale Verfeinerung eines Sprachkonstruktes einer Modellierungs- oder Spezifikationsprache durch ein Fragment einer Implementierungssprache

- ▶ Verfeinerung von Sprachkonstrukten (Realisierung, Abflachen, lowering)
  - Netzentwurf
  - Implementierung von Methoden (von Statecharts und Aktivitätsdiagrammen)
  - Datenverfeinerung
  - Kontrollverfeinerung
  - Syntaktische Verfeinerung
  - Semantische Verfeinerung

# Repräsentation von flexiblen und fixen Objektnetzen als Datenstrukturen (Netzverfeinerung)



## Verständnisfragen

- ▶ Wieso kann man den Klang einer Glocke nicht testen?
- ▶ Wieso braucht man viel Erfahrungen, um die Parameter einer Glocke korrekt zu schätzen?
- ▶ Warum ist das Programmieren von Objektnetzen so schwierig?
- ▶ Welche Möglichkeiten gibt es fürs Testen, wenn man Objektnetz-Bibliotheken verwendet?
- ▶ Warum ist Testautomatisierung für die Programmierung von Objektnetzen so wichtig?



**SUMMARY**