

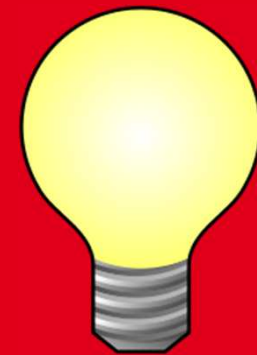


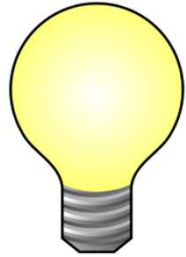
Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# 20.11.2018

## Programmieren im Großen I

Einführung in das Programmieren im Großen





Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# AGENDA

Einführung ins Thema

Der Softwareentwicklungsprozess

Das OO-Vorgehensmodell

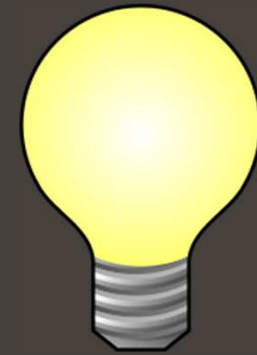
Fazit



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# 01 EINFÜHRUNG INS THEMA

Ziel:  
Die Eckpunkte des Themas kennenlernen



# WORUM GEHT'S?



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Programmieren im Kleinen:
    - kleines Programm ( $\approx$  1 - 20 Klassen)
    - 1 oder 2 Entwickler
    - Schlankes Vorgehen möglich:
      - (Entwurf, Spezifikation,) Implementierung und Test
    - ordentlich (fehlerfrei, wartbar, . . . )
- Kann man auch ohne Prozess oder mit unstrukturiertem Prozess noch einigermaßen gut hinbekommen

# WORUM GEHT'S?



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Programmieren im Großen:
  - Software-System, z.B.:
    - großes Programm (100 Klassen oder mehr)
    - mehrere Programme als Gesamtsystem
  - viele Entwickler
  - echter Kunde(n)
    - Viele Beteiligte (Stakeholder)
  - vollständiger Softwareentwicklungsprozess
  - ordentlich (fehlerfrei, wartbar, . . . )
- Man braucht einen gescheiten Softwareentwicklungsprozess
  - Muss Menschen und Artefakte koordinieren
- Man braucht Projektmanagement, Anforderungen, Design, Implementierung, Testen, ...



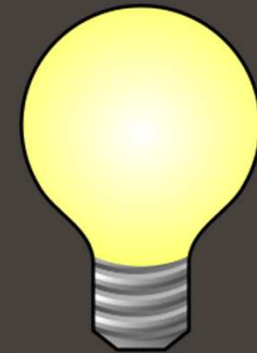
Hochschule RheinMain  
University of Applied Sciences  
Wiesbaden Rüsselsheim

02

## Der Softwareentwicklungsprozess (Wiederholung)

Ziel:

Nochmals den Softwareentwicklungsprozess genauer  
kennenlernen

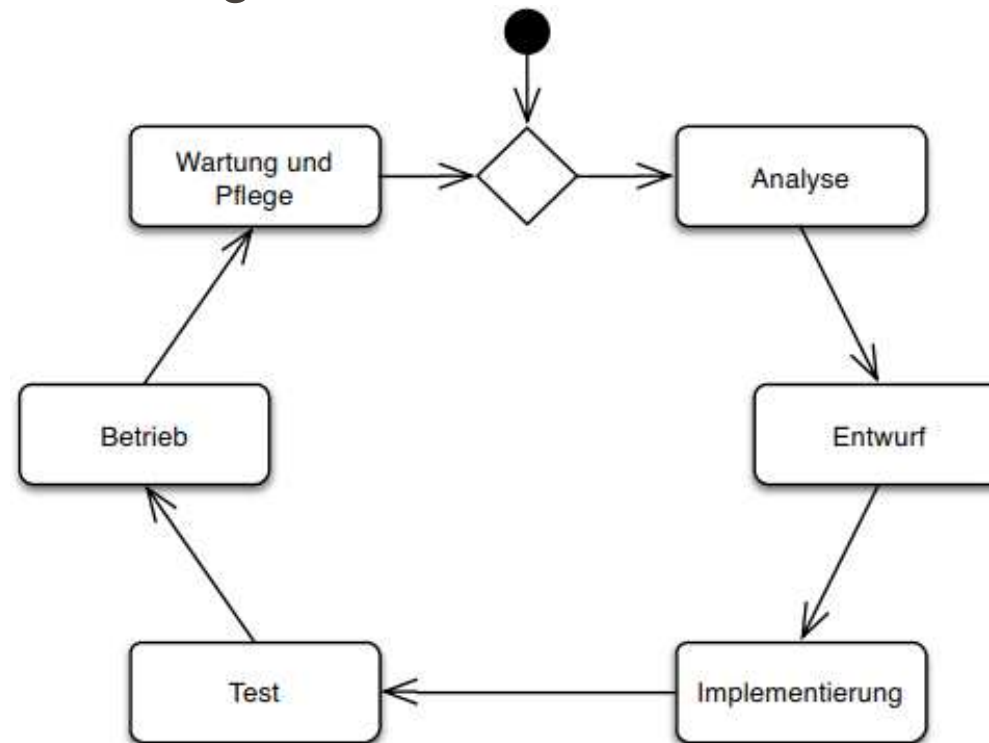


# LEBENSZYKLUS VON SOFTWARE



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

Die typischen Tätigkeiten bei der SW-Entwicklung:



 **Vorsicht: Ist eine Idealisierung!**

→ In der Praxis kann auch mal von Implementierung wieder zur Analyse zurückgesprungen werden, ...

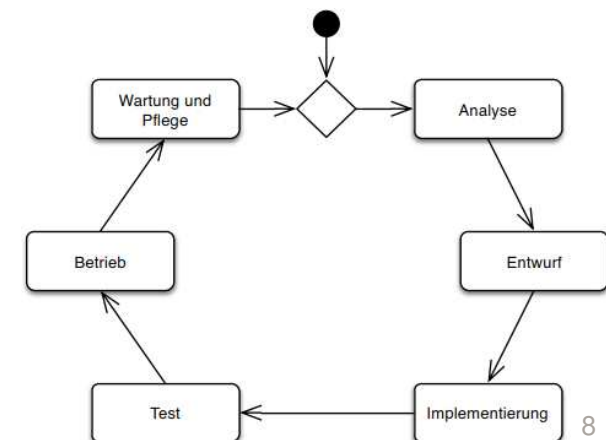
# LEBENSZYKLUS VON SOFTWARE (SOFTWARE-LIFE-CYCLE)



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

Typische Tätigkeiten bei der Software-Entwicklung:

1. Analyse: Was will der Kunde? (= Anforderungen)
2. Entwurf: Wie soll das zu bauende System sein?
  1. grob: Grobentwurf
  2. detailliert: Feinentwurf
3. Implementierung: Entwurf → Programm
4. Test: Erfüllt das Programm die Anforderungen und den Entwurf?
5. Betrieb: Verwendung des Programms
6. Wartung und Pflege
  - Änderungswünsche/Fehler  
→ Was will der Kunde?  
→ . . .





# MEHR VORGABEN SIND NÖTIG



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Typische Fragen bei der Software-Entwicklung:
  - Wie fangen wir an?
  - Was sollen wir tun?
  - Wie verteilen wir die Aufgaben?
  - Wie machen wir's richtig?
  - . . .

→ Hier sind mehr Vorgaben nötig



Hochschule RheinMain  
University of Applied Sciences  
Wiesbaden Rüsselsheim

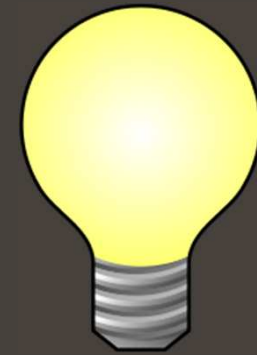
03

## Das OO-Vorgehensmodell

Ziel:

Vorgehensmodelle kennenlernen (Wiederholung)

Unser OO-Vorgehensmodell kennenlernen



# MEHR VORGABEN SIND NÖTIG



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

→ Vorgehensmodelle

= Bestimmte Vorgaben für die Durchführung von Software-Entwicklungs-Projekten

- Typische Vorgaben:
  - Abfolge von Phasen/Tätigkeiten
  - Artefakte = Resultate von Phasen/Tätigkeiten, z.B.
    - Beschreibung der Anforderungen in bestimmter Form
    - Testfallbeschreibungen in bestimmter Form
    - Quellcode-Dateien gemäß Codier-Richtlinien
  - Zusammenhänge zwischen den Phasen/Tätigkeiten
  - Andere organisatorische Aspekte

# VORGEHENSMODELLE – BEISPIELE



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Frühe Modelle
  - Wasserfall
  - V-Modell  
(Deutsche Erfindung – oft benutzt, z.B. Behörden, Automotive)
- Modelle der 2. Generation
  - Spiralmodell (von Barry Boehm)
  - V-Modell mit mehreren Zyklen
- Objektorientierte Modelle (3. Generation)
  - Rational Unified Process (RUP)
- Agile Methoden (4. Generation)
  - eXtreme Programming
  - SCRUM

(Siehe auch Vorlesung 01 und Vorlesung 12 später)

# UNSER OO-VORGEHENSMODELL



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Wir benötigen für das Praktikum, spätere Projekte, . . .
  - einen Rahmen für OOAD (= Objektorient. Analyse & Design)
  - solide
  - erprobt
  - abgespeckte Variante des (R)UP (= Rationale Unified Process)
- RUP insgesamt → für unsere Zwecke viel zu aufwändig
  - Betrachten auszugsweise die für uns wichtigsten RUP-Bestandteile
  - gute Orientierung für die OO-Software-Entwicklung
- Für manche Erklärung verwende ich jedoch auch V-Modell
  - Kann man manches besser erklären
  - Was ich erkläre, ist kompatibel zu RUP (anderes nicht!)

# UNSER OO-VORGEHENSMODELL – AUSGEWÄHLTE TÄTIGKEITEN



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Wir betrachten folgende Tätigkeiten („disciplines“):
  - Anforderungs-Analyse („Requirements“)
    - Was will der Kunde?
  - Analyse und Entwurf („Analysis and Design“)
    - Wie soll das zu bauende System sein?
  - Implementierung („Implementation“)
    - Das System bauen
  - Test:
    - Wie stelle ich sicher, dass das System das tut, was es tun soll?
- Wir gehen nicht ein auf:
  - Geschäftsprozessmodellierung („Business Modeling“)
  - Inbetriebnahme („Deployment“)

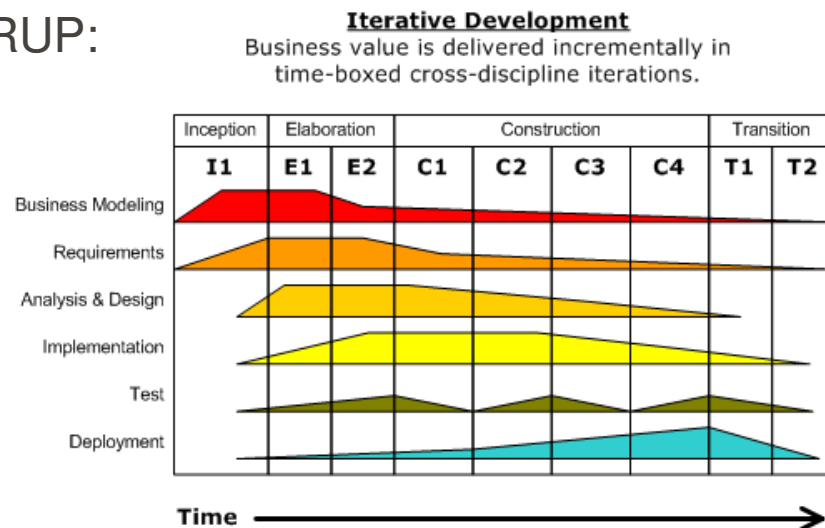
# UNSER OO-VORGEHENSMODELL – AUSGEWÄHLTE TÄTIGKEITEN



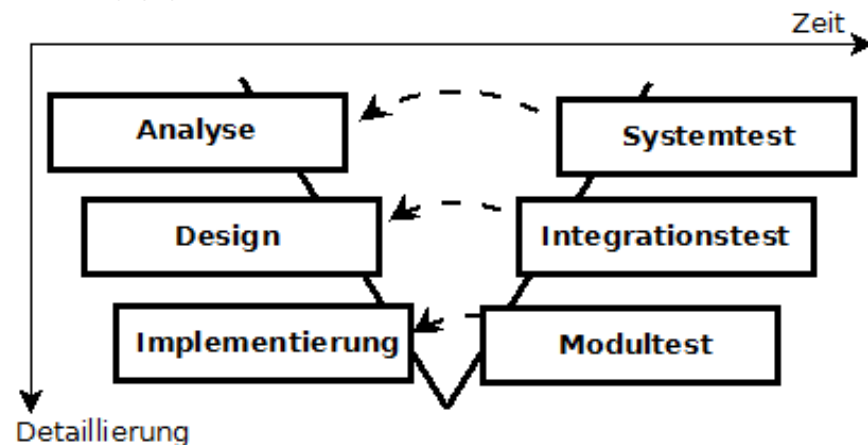
Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Wir betrachten folgende Tätigkeiten („disciplines“):
  - Anforderungs-Analyse („Requirements“)
  - Analyse und Entwurf („Analysis and Design“)
  - Implementierung („Implementation“)
  - Test

RUP:



V-Modell:



→ Ich verwende die V-Modellzeichnung (eingängiger für Sie), aber ich erkläre die Tätigkeiten nach RUP-Stil (etwas besser an OO angelehnt)



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

## 04 Fazit

Ziel:  
Was haben wir damit gewonnen?







# WAS HABEN WIR GELERNT?

- Wofür steht Programmieren im Großen?
  - Entwicklung großer SW-Systeme
    - Mehrere Programme, viele Klassen, ...
  - Viele Beteiligte
- Zur Koord. braucht man einen SW-Entwicklungsprozess
  - Es gibt hier viele verschiedene
    - Sog. Vorgehensmodelle
- Unser OO-Vorgehensmodell:
  - Abgespecktes RUP-Vorgehensmodell
- Zur leichteren illustration verwende ich aber V-Modell

# WEITERFÜHRENDE LITERATUR



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Kleuker: Grundkurs Software-Engineering mit UML  
[<http://dx.doi.org/10.1007/978-3-8348-9843-2>].
- Zuser et al: Software-Engineering mit UML und dem Unified Process [BF 500 92].
- C. Larman: Applying UML and Patterns [30 BF 500 78].



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

AUF GEHT'S!!

SELBER MACHEN UND LERNEN!!

