

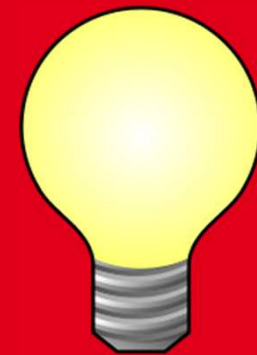


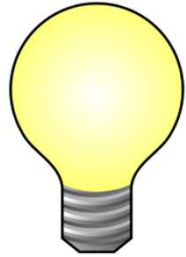
Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

23.10.2018

# Softwaretechnik – Was ist das?

Einführung in die Softwaretechnik-Vorlesung





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

# AGENDA

Einführung ins Thema

Was ist Softwaretechnik?

Der Softwareentwicklungszyklus

Vorgehensmodelle

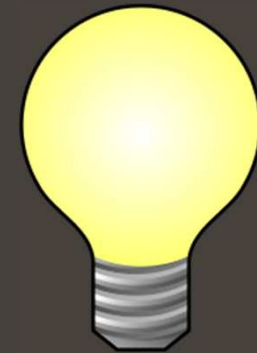
Fazit



Hochschule RheinMain  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# 01 EINFÜHRUNG INS THEMA

Ziel:  
Die Eckpunkte des Themas kennenlernen



# GRUNDLEGENDE FRAGEN ZU SWT



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

Sie werden sich zu Softwaretechnik fragen:

- Was ist das?
- Wozu braucht man das?

→ Siehe Film „clip\_1\_final.mp4“



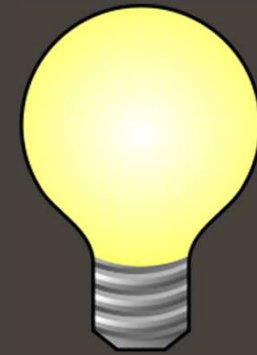
Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

02

## Was ist Softwaretechnik?

Ziel:

Was versteht man unter Softwaretechnik?



# AUSGANGSPUNKT



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

## Typische Probleme bei der Software-Entwicklung:

- viele IT-Projekte scheitern
- Konsequenzen fehlerhafter Software
- Kosten und Dauer von SW-Entwicklungsprojekten geraten aus dem Ruder
- . . .

# ZIEL:



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

→ Günstige Software hoher Qualität

- Hohe Qualität:
  - fehlerfrei
  - schnell
  - schnell/leicht veränderbar
  - leicht testbar
  - benutzerfreundlich
  - sicher

# ANSATZ I: INDUSTRIALISIERUNG



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

- Entwicklungsstufen:
  - Laie → Zufallstreffer
  - Handwerker → teure Einzelanfertigung
  - Ingenieur → Qualität am Fließband
- Software-Entwicklung als Ingenieursdisziplin:
  - Softwaretechnik (Software Engineering)
- Erfolgsfaktoren:
  - erprobte Methoden
  - erprobte Werkzeuge



# DEFINITIONEN



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

- Softwaretechnik (Software Engineering) =
  - Methoden + Werkzeuge + Hilfsmittel,
  - die Softwareentwickler dabei unterstützen,
  - für Geld reproduzierbar
  - Software hoher Qualität herzustellen.
  
- Software =
  - Programm
  - Konfigurationsdateien
  - Dokumentation
  - . . .

# HINWEISE ZU DEN BEGRIFFEN



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

- Ich werde auf öfters Software Engineering benutzen
  - Ist der übliche / etablierte Begriff
  - Auch im deutschsprachigen Raum
- Abkürzung: SE

Weiterer Hinweis: Es gibt auch Systems Engineering

- Hier geht es darum ganze Systeme zu entwickeln
  - HW, Software, Mechanik, Elektrotechnik, Chemie, ...
- Auch hier ist Software involviert
- Diesen Aspekt klammern wir hier aus
  - Hier geht es nur um reine Softwaresysteme
  - Näheres zu SysEng können Sie in der Anforderungsmanagement-Vorlesung erfahren

# ANSATZ II: SOFTWARE CRAFTMANSHIP



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

## Ansatz I – Industrialisierung (traditionell)

- SE als Ingenieursdisziplin, die zur „Industrialisierung“ führt
- Z.B. durch Strukturierte Methoden
- Große Fortschritte, aber nicht so durchschlagende Erfolge wie ursprünglich erwartet

## Ansatz II – Meisterliches Handwerk (Software Craftmanship)

- Man ist bescheidener geworden
- SW ist anders als z.B. Bauingenieurwesen
  - SW ist z.B. sehr abstrakt und es sind andere Lösungen mögl.
- Agile Methoden

# ANSATZ II: SOFTWARE CRAFTMANSHIP



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

- Ansatz I – Industrialisierung (traditionell)
  - Ansatz II – Meisterliches Handwerk (Software Craftmanship)  
→ z.B.: Agile Methoden
  - Erfolgsfaktoren für Ansatz II:
    - erprobte Methoden
    - erprobte Werkzeuge
- } NUR: Etwas andere Methoden und Werkzeuge

## Was lernen wir hier?

- Hauptsächlich Ansatz I (evtl. ein paar Bemerk. zu Ansatz II)
- Ansatz I ist nicht verkehrt → wird noch sehr oft benutzt
- Bildet die absolute Basis für weiterführende Überlegungen
- ABER: Es gibt eben wesentlich mehr als wir hier lernen können

# KERNTHEMEN FÜR UNS



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

Das sind für uns die wichtigsten Themen in diesem Semester:

- Strukturierte Vorgehensweise: hier OOAD  
(Object-oriented Analysis and Design)
- einheitliche Notation für SW-Modelle
  - Hier: UML (Unified Modeling Language)
  - FMC (Fundamental Modeling Concepts)
- Konzepte und Abstraktionen: z.B. Muster, SW-Architektur
- Qualitätssicherung – z.B. Testen
- CASE (Computer Aided Software Engineering)

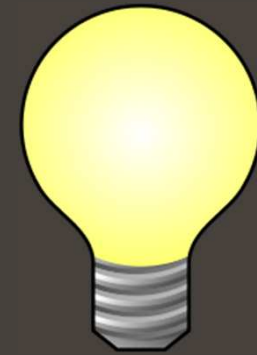


Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# 03

## Der Softwareentwicklungszyklus

Ziel:  
Den grundlegenden Entwicklungszyklus  
für Software kennenlernen



# WICHTIGE BEGRIFFE – ARTEFAKT



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

- Beschreibt -ganz abstrakt- ein von Menschen künstlich (artifiziert) erstelltes Objekt.
- Ein Artefakt ist nicht notwendigerweise ein Dokument
  - Ein Art. kann durch eines oder mehrere Dok. ausgedrückt werden
  - Ein Dok. kann aber auch mehrere Art. beinhalten

## Beispiel:

- Das Artefakt „Quellcode“ besteht meist aus mehreren Codefiles (=mehrere Dokumente).
- Gut sich selbst dokumentierender Code (z.B. dokumentierte Methodenköpfe) enthält auch schon Teile des Artefaktes „Dokumentation“.
  - Können (z.B. über JavaDoc) noch in einen explizites Dokumentationsdokument transformiert werden.

# WICHTIGE BEGRIFFE – STAKEHOLDER



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

- Englisch für Interessenvertreter, Akteur
  - „Stabhalter“ - Staffellauf
  - Something is at stake == etwas steht auf dem Spiel

**Def nach Pohl und Rupp [PR15; S.4]:**

Ein Stakeholder eines Systems ist:

- eine Person oder Organisation, die
- direkten oder indirekten Einfluss auf das Projekt (v.a. die Anforderungen) des betrachteten Systems hat.



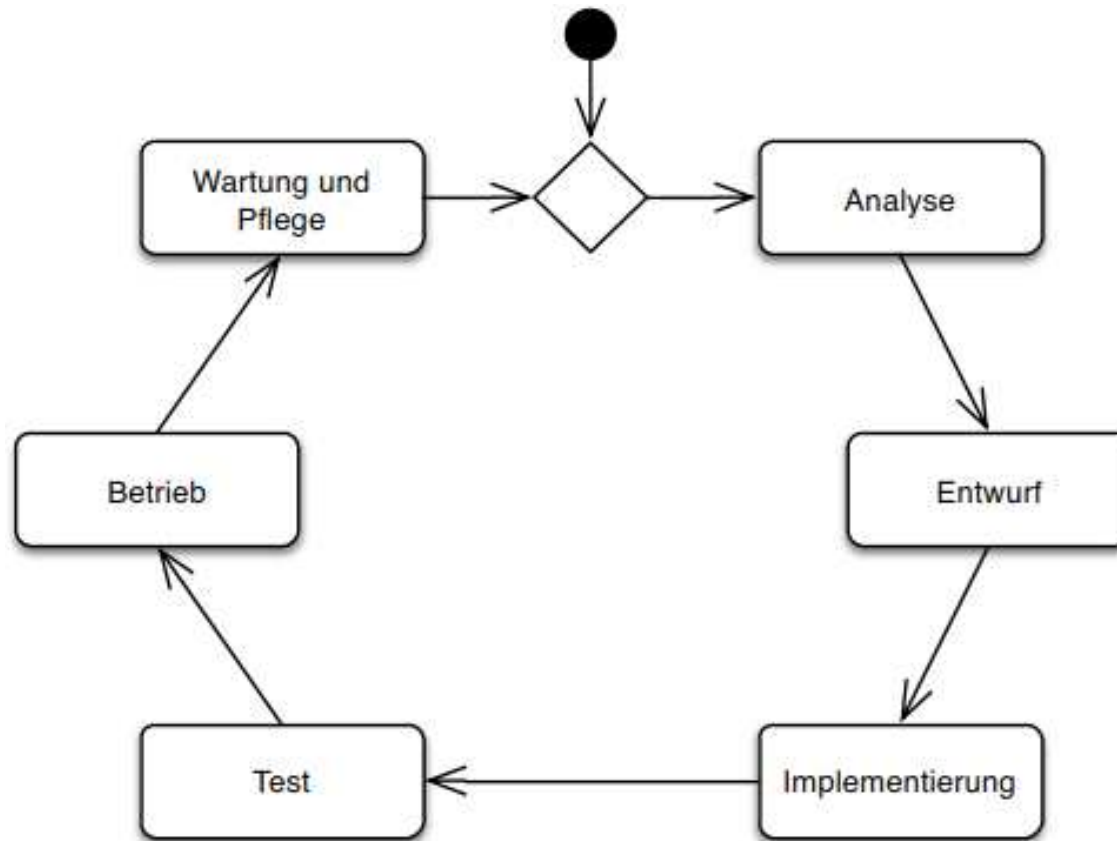


# LEBENSZYKLUS VON SOFTWARE



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

→ Beschreibt die typischen Tätigkeiten bei der SW-Entwicklung



# LEBENSZYKLUS VON SOFTWARE



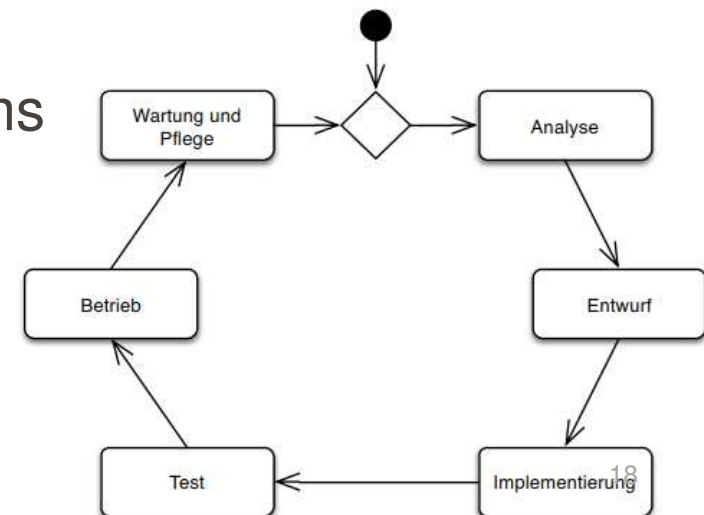
Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

→ Beschreibt die typischen Tätigkeiten bei der SW-Entwicklung

1. Analyse: Was will der Kunde? (= Anforderungen)
2. Entwurf: Wie soll das zu bauende System sein?
  - grob: Grobentwurf
  - 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?

→ . . .

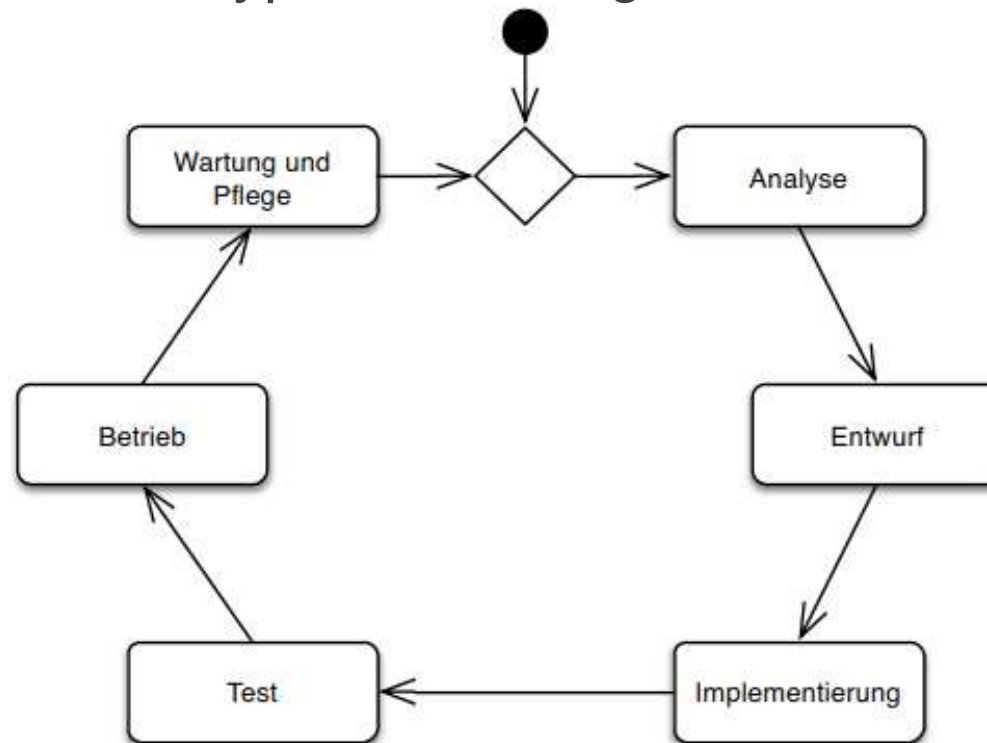


# LEBENSZYKLUS VON SOFTWARE



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

→ Beschreibt die typischen Tätigkeiten bei der SW-Entwicklung



 **Vorsicht: Ist eine Idealisierung!**

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

# 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

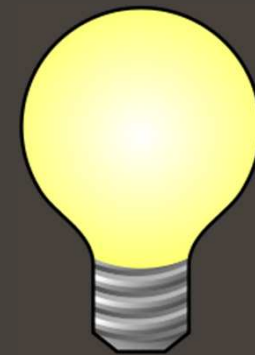


Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

# 04

## Vorgehensmodelle

Ziel:  
Vorgehensmodelle 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



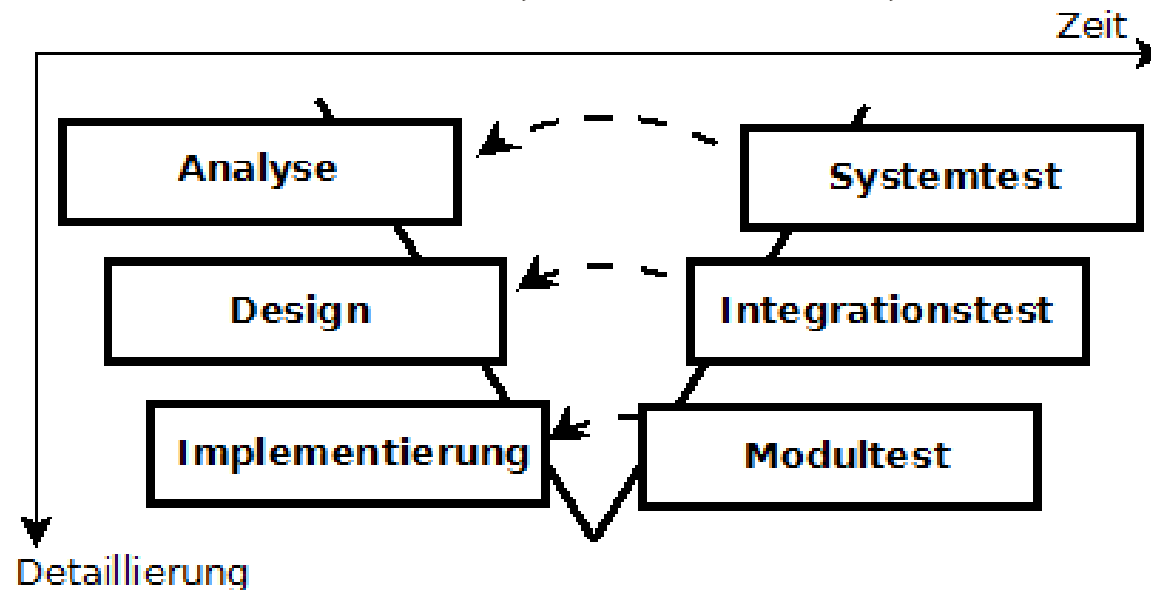
Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

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



# V-MODELL

- Abwandlung des Wasserfall-Modells
  - Deutsche Erfindung (TU München → siehe Manfred Broy)
  - oft in Deutschland benutzt, z.B. Behörden, Automotive, ...



- Leider sehr starr
- Zunächst nicht iterativ geplant → Nur ein Zyklus

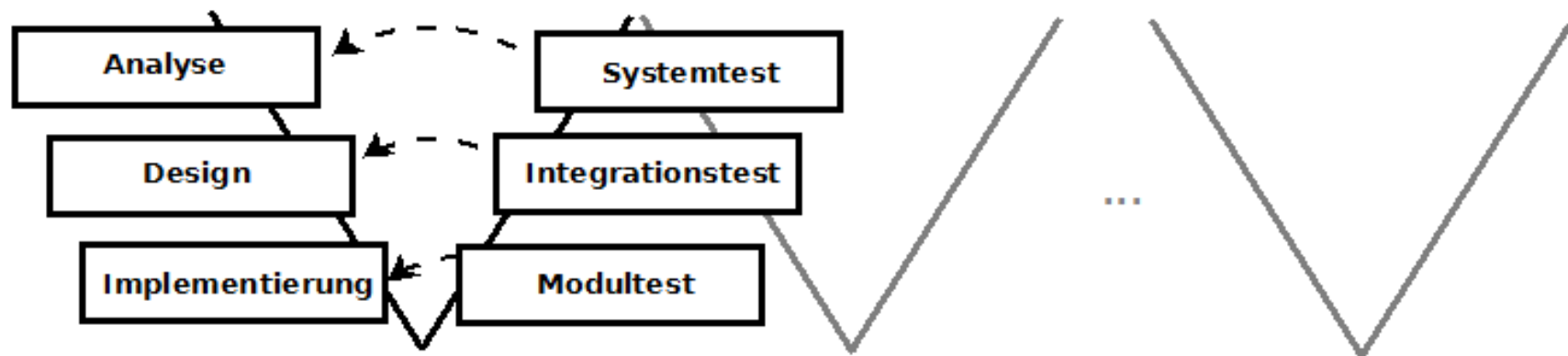


# V-MODELL



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

- Weiterentwicklung für iterative Entwicklung:



- Redesign (seit 2005): V-Modell XT
  - Iterativ, inkrementell, wesentlich flexibler und besser skalierbar (Reaktion auf RUP, Spiralmodell & Agile Methoden)



# SPIRALMODELL

- Entwickelt von Barry Boehm
  - Iterativ



Bildquelle: „Spiralmodel nach Boehm.png“ von WikiMedia Commons

# RATIONAL UNIFIED PROCESS (RUP)

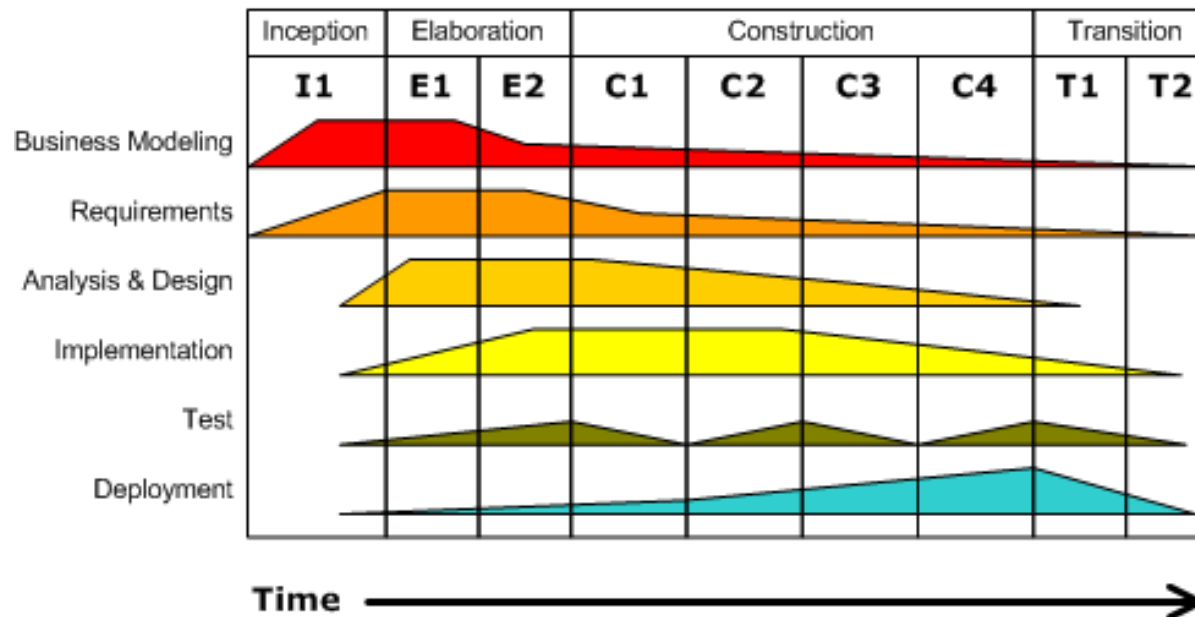


Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- Entwickelt parallel zur UML
  - Von der Firma Rational (jetzt IBM)

## Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



- Quelle: „Development-iterative.png“ von Wikimedia Commons

# AGILE METHODEN



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

- Relativ neu
  - Abkehr von starren Prozessen
  - Keine richtigen Vorgehensmodelle in klassischen Sinn
  - Eher Sammlung von Prinzipien (Best Practices)
    - Muster-Idee → Prozessmuster
    - (siehe Vorlesungseinheit zu Mustern)
- Typische Beispiele:
  - eXtreme Programming (XP) → Für Entwicklung
    - Nach Win Vista-Katastrophe hat Microsoft auf XP gesetzt → Win 7
  - SCRUM
    - Eigentlich eher eine Projektmanagementmethode
    - Für Entwicklung kann z.B. XP genutzt werden oder anderes
    - Derzeit richtig „IN“

# WAS MACHEN WIR JETZT



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

- Das war jetzt nur mal zur Orientierung
  - In Vorlesung 12\_ProgrammierenImGrossen\_VI\_Durchfuehren\_von\_Projekten werden wir darauf nochmals genauer zurückkommen
- Woran sollten Sie sich jetzt orientieren?
  - Wir benötigen für das Praktikum, spätere Projekte,
    - einen Rahmen für OOAD
    - solide
    - erprobt
  - V-Modell und RUP
  - V-Modell, weil es sehr eingängig von den Phasen her ist
  - RUP, weil es für Objektorientierte Entwicklung spezielle viele gute Sachen vorstellt
  - Beides ist dazu sehr kompatibel

# WELCHE TÄTIGKEITEN BETRACHTEN WIR?



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

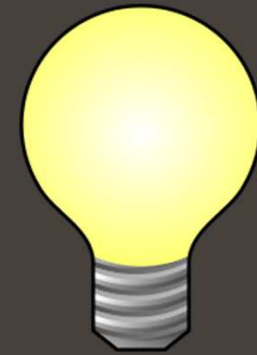
- Wir betrachten folgende Tätigkeiten (RUP: „disciplines“)
  - Anforderungs-Analyse (RUP: „Requirements“)
    - Was will der Kunde?
  - Analyse und Entwurf (RUP: „Analysis and Design“)
    - Wie soll das zu bauende System sein?
  - Implementierung (RUP: „Implementation“)
    - Das System programmieren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



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

# 05 Fazit

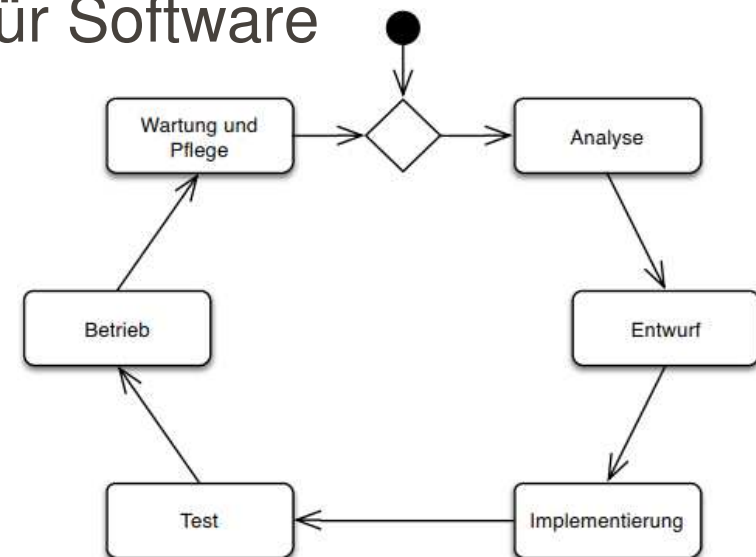
Ziel:  
Was haben wir damit gewonnen?





# WAS HABEN WIR GELERNT?

- Den Begriff Software Engineering kennenlernen
  - Lehre von Methoden + Werkzeugen + Hilfsmitteln zur systematischen Entwicklung von Software hoher Qualität
- Grundlegende Entwicklungszyklus für Software
- Vorgehensmodelle
  - Weiteres, genaueres Vorschriftengerüst





# WORUM GEHT ES IN DEN KOMMENDEN VORLESUNGEN?



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

- Modellierung von Software
  - Mittels Zeichnungen Eigenschaften einer Software herausarbeiten
- Wir lernen die Modellierungssprache UML kennen



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

AUF GEHT'S!!

SELBER MACHEN UND LERNEN!!

