

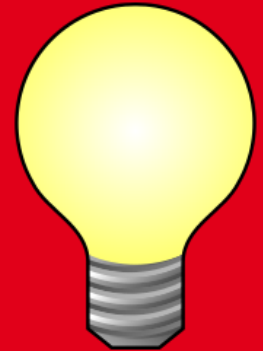


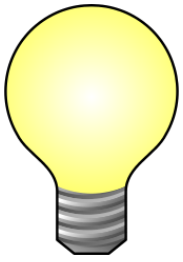
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

07.01.2021

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

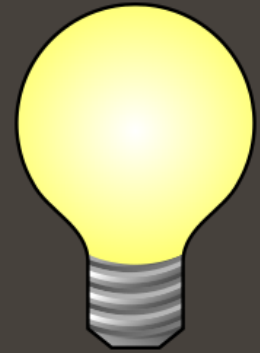
Fazit



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

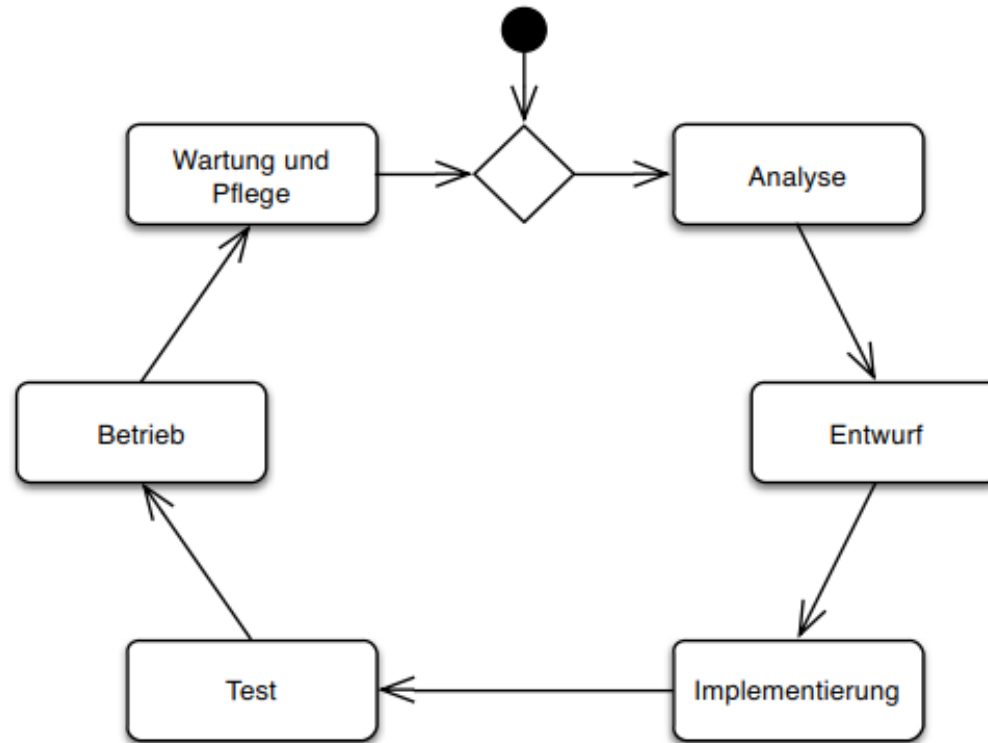


LEBENSZYKLUS VON SOFTWARE



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, ...

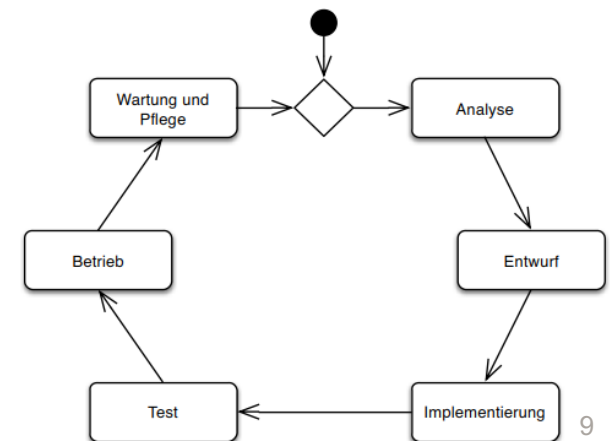
LEBENSZYKLUS VON SOFTWARE (SOFTWARE-LIFE-CYCLE)



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 (Architektur/Architecture)
 2. detailliert: Feinentwurf (Detailed Design)
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 Softwareentwicklungsprojekten

- 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-Softwareentwicklung
- 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“):
 - Anforderungsanalyse („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



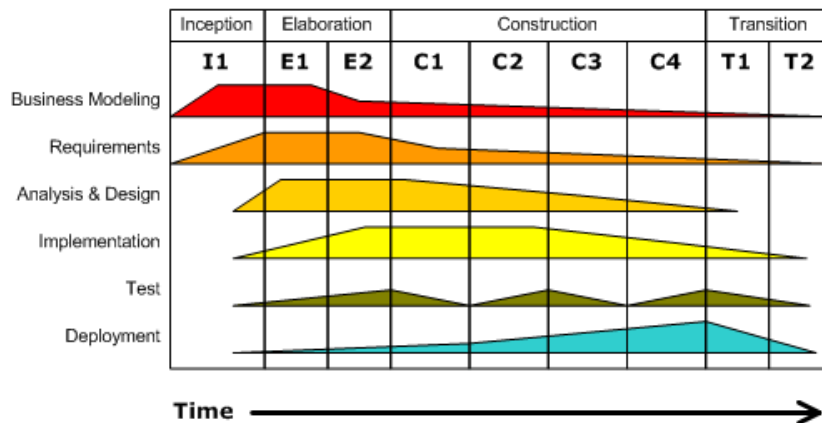
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

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

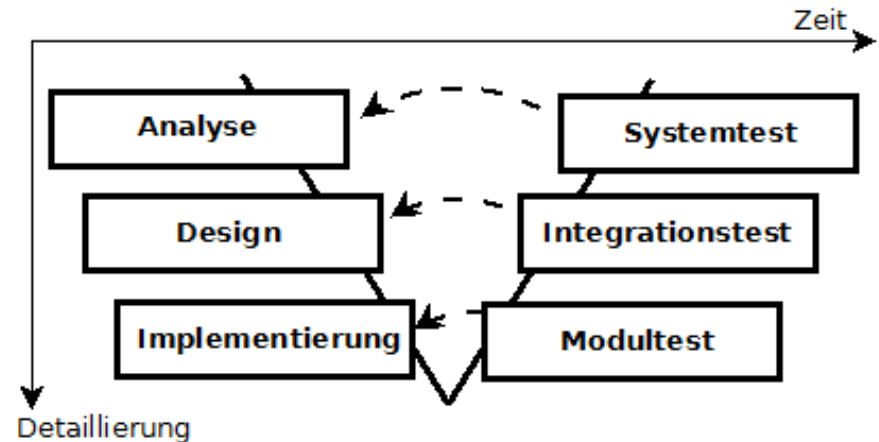
RUP:

Iterative Development

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



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 Koordin. 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].
- Larman, C.: Applying UML and Patterns [30 BF 500 78].



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

AUF GEHT'S!!

SELBER MACHEN UND LERNEN!!



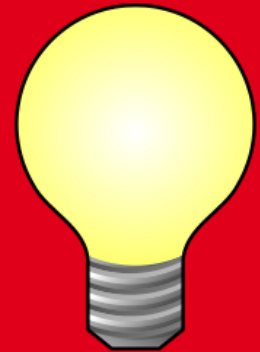


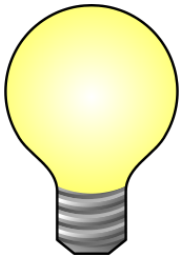
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

07.01.2021

Programmieren im Großen II

Anforderungsanalyse





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

AGENDA

Einführung ins Thema

Das Fachmodell

Anwendungsfälle (Use Cases)

Nichtfunktionale Anforderungen

GUI

Qualitätssicherung

Das war's noch lange nicht

Fazit



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

01 EINFÜHRUNG INS THEMA

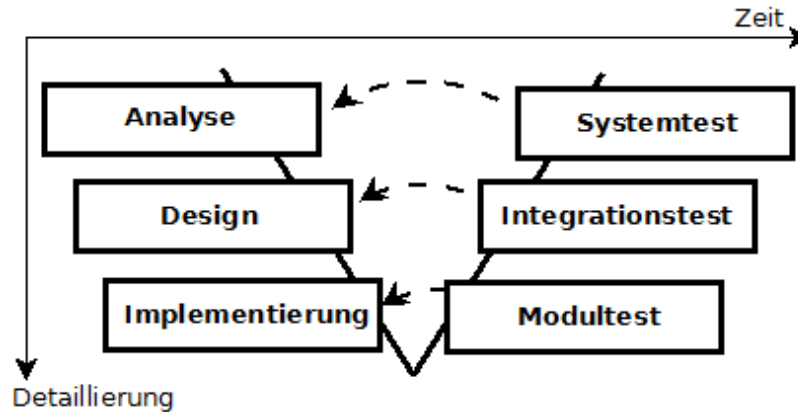
Ziel:
Die Eckpunkte des Themas kennenlernen



WORUM GEHT'S?



- Letztes Mal:
 - Wir brauchen ein Vorgehensmodell
 - Verschiedene Phasen:

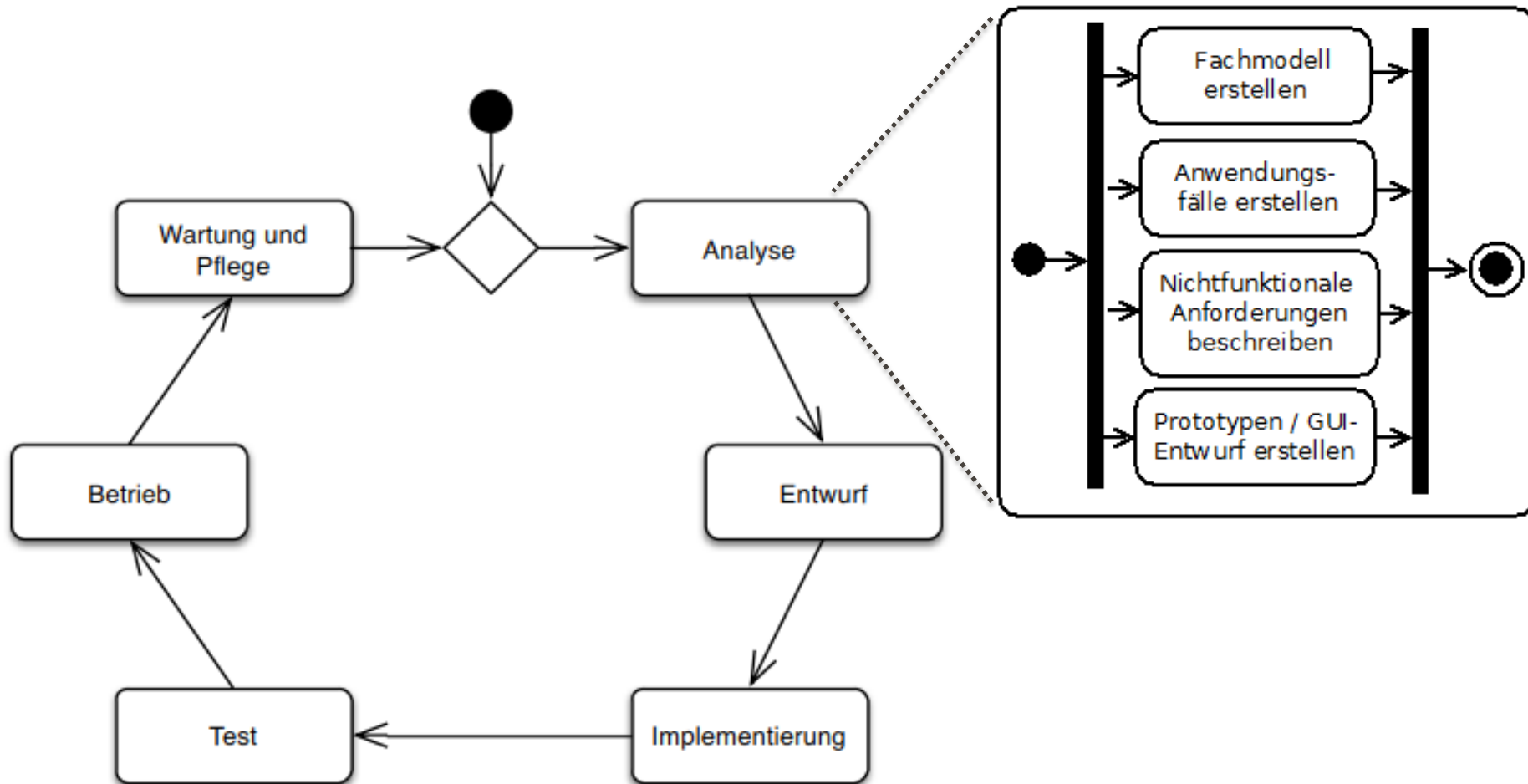


- Heute:
 - Anforderungsanalyse

ANFORDERUNGSANALYSE – UNTERAKTIVITÄTEN:



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



ANFORDERUNGSANALYSE – EINFÜHRUNG



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

- Ziel:
 - Wir wissen genau, was der Kunde will
- Input: Kundengespräche, vorhandene Dokumente, . . .
- Output:
 - statische Aspekte: „Analyse-Klassen-Diagramm“ = „Fachmodell“
= im wesentlichen: UML-Klassen-Diagramm(e)
(„Datenmodell“, „Domänenmodell“ („domain model“))
 - dynamische Aspekte: Anwendungsfälle („Use Cases“)
= im wesentlichen textuelle Beschreibungen
 - Nichtfunktionale Anforderungen (z.B. Performance, Sicherheit,...)
 - ggfs. Prototypen, GUI-Entwurf, . . .

DURCHGÄNGIGES BEISPIEL – MP3-PLAYER



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

- Für die Erklärung der Phasen nutze ich möglichst ein durchgängiges Beispiel
 - Angenommen wir sollen einen MP3-Player für den PC (wie z.B. WinAmp) entwickeln
- BEM: Ist vielleicht etwas veraltet
(nicht so sexy, wie eine Handy-App)
 - ABER:
 - Sehr einfaches Beispiel
 - Sehr eingängig (jeder von Ihnen kennt das)
→ MP3-Player auf dem Handy funktionieren ähnlich



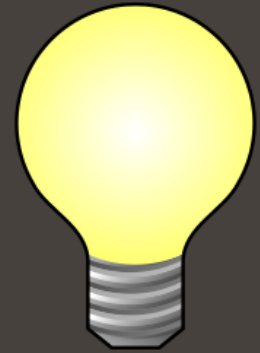
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

02

Das Fachmodell

Ziel:

Das Fachmodell kennenlernen

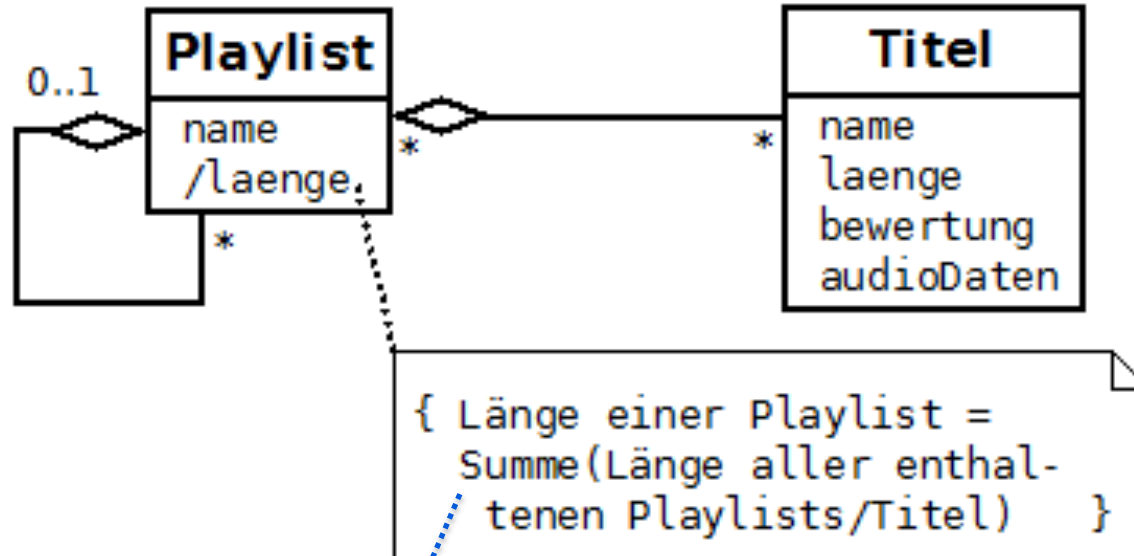


FACHMODELL – BEISPIEL



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

- Beispiel: (unvollständiges) Fachmodell für die Playlistenverwaltung eines Software-MP3-Players:



Einschränkung (Constraint)
mit {...} formuliert

FACHMODELL – DEFINITION & FORM



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

- Definition Fachmodell:
 - Beschreibung der statischen Aspekte (Daten) des Anwendungsgebiets in der Sprache des Anwendungsgebiets
(= Sprache des Kunden)
- Form:
 - UML-Klassendiagramm(e)
 - + Kommentare
 - + Einschränkungen (Constraints)
 - (in natürlicher oder formaler Sprache (z.B. OCL = Object Constraint Language))
 - dynamische Aspekte (d.h. keine Methoden und Interfaces)
 - Auch Attribut-Typen und andere Details dürfen weggelassen werden
 - BEM: Auch ER-Modelle eignen sich natürlich vorzüglich

FACHMODELL – WAS IST ES NICHT?



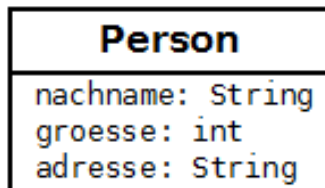
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim



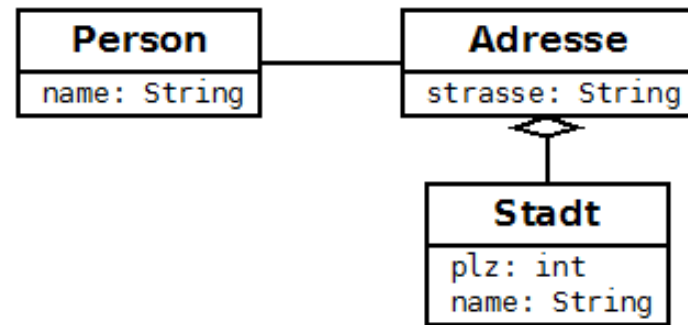
Vorsicht

- i.d.R.: Klassen-Diagramm für Fachmodell \neq Klassen-Diagramm für Java-Klassen
- gleiche Notation für völlig verschiedene Anwendungsbereiche!
- Fachmodell bildet nur die fachlichen Zusammenhänge ab
- Im Design kommen noch viele andere Aspekte hinzu
 - Fachmodell bildet aber meist die Ausgangsbasis

Fachmodell:



“Fachmodell” im Design:



→ In Zukunft Domänenmodell
genannt!

FACHMODELL – WIE KOMMT MAN DAZU?



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

- Bewährte Vorgehensweisen
 - Bewährtes aus der Vorlesung „Datenbanken“
 - z.B.: Normalformen
 - Analysemuster (→ siehe Praktikum)
- Wie finde ich die Klassen, Attribute, Assoziationen, . . . für das Fachmodell?
 - Hilfsmittel I: Typische Situationen mit Hilfe von Objektdiagrammen veranschaulichen → OD zeichnen
 - Hilfsmittel II: Textanalyse nach R. J. Abbott: „Program Design by Informal English Descriptions“. Comm. of the ACM, 26, #11, 1983.
 - Hilfsmittel III: Methode von Bjarne Stroustrup
 - ... es gibt noch andere Hilfsmittel
 - TIP: Am besten miteinander kombinieren

TEXTANALYSE NACH ABOTT* - VORGEHENSWEISE:



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

1. Gut zuhören & relevante Daten sammeln
2. Zuordnung:
 - Hauptwörter (Substantive) → Liste von Objekten, Attributen
 - Zeitwörter (Verben) → Liste von Beziehungen (& Methoden)
3. Liste bereinigen:
 - Redundanzen streichen (z.B. Synonyme)
 - Irrelevante Begriffe streichen
 - Vage Begriffe streichen oder präzisieren
4. Klassen mit Attributen und Beziehungen (Assoziationen) ableiten → Klassendiagramm erstellen

* R. J. Abbott: „Program Design by Informal English Descriptions“. Comm. of the ACM, 26, #11, 1983.

B. STROUSTRUP** – BEZIEH. ZW. KLASSEN FINDEN*



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

<i>Ist</i> a ein b?	Ist b ein Spezialfall von a? Ein Auto <i>ist</i> ein Fortbewegungsmittel.
<i>Hat</i> a ein b?	<i>Besitzt</i> a ein Objekt vom Typ b? Ein Auto <i>hat</i> einen Lenker. Ein Auto <i>ist aber kein</i> Lenker.
<i>Benutzt</i> a ein b?	<i>Benutzt</i> a ein Objekt vom Typ b? Ein Auto <i>benutzt</i> Straßen. Ein Auto <i>ist keine</i> Straße. Ein Auto <i>hat keine</i> Straße.

* Siehe auch OOSE: 14_OO5_Wie_zum_Objektmodell

** B. Stroustrup: The C++ Programming Language, 1998.

B. STROUSTRUP** – BEZIEH. ZW. KLASSEN FINDEN*



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

Deutet auf Vererbung hin



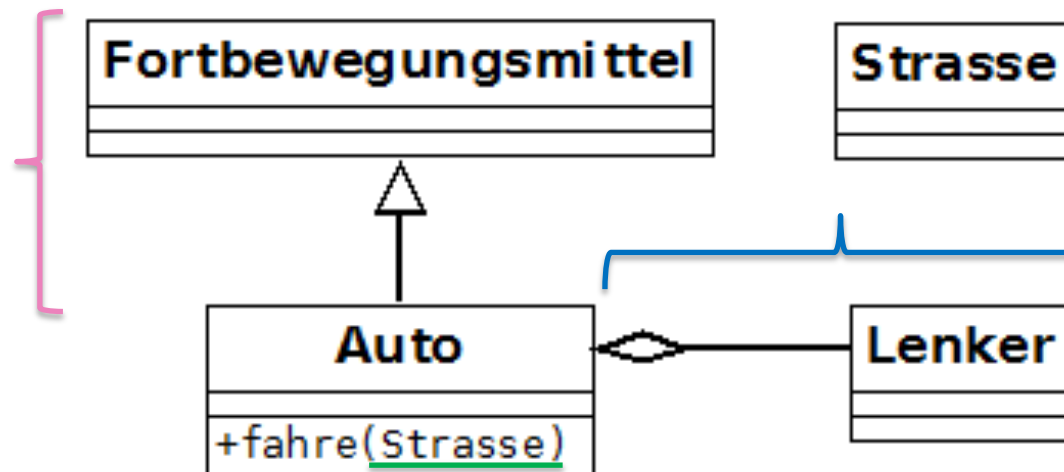
Deutet auf Attribut hin



Deutet auf anderes hin
z.B.: Parameter in Methode,
Attribut, das benutzt wird,
Assoziation, ...



<i>Ist</i> a ein b?	Ist b ein Spezialfall von a? Ein Auto <i>ist</i> ein Fortbewegungsmittel.
<i>Hat</i> a ein b?	<i>Besitzt</i> a ein Objekt vom Typ b? Ein Auto <i>hat</i> einen Lenker. Ein Auto <i>ist aber kein</i> Lenker.
<i>Benutzt</i> a ein b?	<i>Benutzt</i> a ein Objekt vom Typ b? Ein Auto <i>benutzt</i> Straßen. Ein Auto <i>ist keine</i> Straße. Ein Auto <i>hat keine</i> Straße.



* Siehe auch OOSE: 14_OO5_Wie_zum_Objektmodell

** B. Stroustrup: The C++ Programming Language, 1998.

TIP: ANSÄTZE AM BESTEN KOMBINIEREN



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

Mögliche Kombination der Ansätze:

1. Textanalyse nach Abott

- Mögliche Klassenkandidaten, Attribute, Aktionen, ... finden

2. Beziehungen nach B. Stroustrup analysieren

- Beziehungen zwischen Klassenkandidaten, ... aufdecken

3. Im Zweifelsfalls / wenn es nicht klar ist?

→ Objektdiagramme verschiedener Situationen zeichnen

+ Evtl. andere Analysemuster

(siehe spätere Vorlesung zu Mustern & Praktikum)



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

03

Anwendungsfälle (Use Cases)

Ziel:

Anwendungsfälle spezifizieren lernen

→ Spezifikation der Funktionalen Anforderungen



ANWENDUNGSFÄLLE (USE CASES)



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

- Definition Anwendungsfälle (Use Cases):
 - Beschreibung der dynamischen Aspekte (**Datenverarbeitung**) des Anwendungsgebiets in der Sprache des Anwendungsgebiets (= Sprache des Kunden)
- Definition Anwendungsfall (Use Case):
 - **textuelle** Beschreibung **eines** Vorgangs eines Systems

ANWENDUNGSFÄLLE – ANWENDUNGSFALLSCHABLONE



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

- Wie geht man jetzt genau vor?
 - Unterstützung durch Schablone (Template) \triangleq Formular
 - Die einzelnen Punkte unterstützen dabei, um nichts wesentliches zu vergessen
- Wo gibt es Anwendungsfallschablonen?
 - z.B. auf <http://alistair.cockburn.us>
 - z.B. Seite der Sophist Group: <https://www.sophist.de>



Vorsicht:

- Eine Schablone ist nur eine Schablone.
- Verstehen & an die eigenen Bedürfnisse anpassen!

ANWENDUNGSFÄLLE – ANWENDUNGSFALLSCHABLONE



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

Vorschlag was in Schablone vorkommen sollte:

- Name + Eindeutig ID (macht Querverweise leichter)
- Ziel
- Beteiligte Akteure
- Verwendete (andere) Anwendungsfälle
- Auslöser, Vorbedingung, Nachbedingung
(für Erfolgs- und Misserfolgs-Fälle)
- Standardablauf (Szenario, in dem alles gut und normal verläuft)
- Alternative Ablaufschritte
 - Auflistung **aller** Ausnahmen/Fehlerfälle
 - Verhalten in Ausnahmesituation
- Zeitverhalten, Verfügbarkeit
- Fragen & Kommentare

Querverweise

Nichtfunktionale
Anforderungen

ANWENDUNGSFÄLLE – ANWENDUNGSFALLSCHABLONE



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

Am besten man verwendet eine Tabelle:

ID: Name	
Ziel	
Akteure	
Status	
Verwendete UseCases	
Auslöser	
Vorbedingungen	
Nachbedingungen (Ergebnis)	Erfolg: Misserfolg:
Standardablauf	
Alternative Ablaufschritte	
Zeitverhalten	
Verfügbarkeit	
Fragen	
Kommentare	

ANWENDUNGSFALLBESCHREIB. – BEISPIEL (MP3-PLAYER):




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

ID: Name	UC001: Titel hinzufügen
Ziel	Der Titel ist in der gewünschten Playliste enthalten.
Akteure	Anwender
Status	Entwurf
Verwendete UseCases	-
Auslöser	Kauf/Erhalt eines neuen Titels.
Vorbedingungen	Titel muss in Form einer Audio-Datei im Datei-System vorliegen. Das System ist geöffnet und zeigt das Hauptfenster
Nachbedingungen (Ergebnis)	Erfolg: Keine Veränderung an der Original-Audio-Datei, System enthält die Informationen über den Titel und die zugehörigen Audio-Daten. Misserfolg: Keine Veränderung am System.

ANWENDUNGSFALLBESCHREIB. – BEISPIEL (MP3-PLAYER):



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

Standardablauf	<p>1: Anwender klickt auf die Funktion „Titel hinzufügen“.</p> <p>2: Das System zeigt den Dialog „Titel hinzufügen“ (DIALOG_XY)</p> <p>3: Anwender wählt eine Playlist aus.</p> <p>4: ...</p> <p>5: ...</p> <p>6: ...</p> <p>7: System importiert den Titel.</p>	 <p>Querverweis über ID auf einen GUI-Entwurf (siehe später)</p>
Alternative Ablaufschritte	<p>3a: Noch keine Playlist vorhanden</p> <p>...</p> <p>4a: Titel bereits vorhanden</p> <p>4b: ... andere Ausnahme ...</p> <p>...</p> <p>7a: Import schlägt fehl</p> <p>...</p>	
Zeitverhalten	Keine Einschränkung	
Verfügbarkeit	Muss immer möglich sein	
Fragen		
Kommentare		

→ 1. Iteration

ANWENDUNGSFALLBESCHREIB. – BEISPIEL (MP3-PLAYER):



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

Standardablauf	1: Anwender klickt auf die Funktion „Titel hinzufügen“. 2: Das System zeigt den Dialog „Titel hinzufügen“ (DIALOG_XY) 3: Anwender wählt eine Playlist aus. 4: ... 5: ... 6: ... 7: System importiert den Titel.
Alternative Ablaufschritte	3a: Noch keine Playlist vorhanden ... 4a: Titel bereits vorhanden
<div data-bbox="154 818 419 911">Alternative Ablaufschritte</div> <div data-bbox="48 1003 150 1046">Zeitv</div> <div data-bbox="48 1061 150 1103">Verf</div> <div data-bbox="48 1118 150 1160">Frag</div> <div data-bbox="48 1175 150 1218">Kom</div>	3a: Noch keine Playlist vorhanden → System erstellt automatisch eine Default-Playlist ... 4a: Titel bereits vorhanden → Korrektur durch Anwender oder Abbruch 4b: ... andere Ausnahme ... → 7a: Import schlägt fehl → ??? ...

Das ist noch sehr ungenau
→ Reaktionen fehlen

→ 2. Iteration

ANWENDUNGSFALLBESCHREIB. – BEISPIEL (MP3-PLAYER):



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

Standardablauf	1: Anwender klickt auf die Funktion „Titel hinzufügen“. 2: Das S 3: Anwe 4: ... 5: ... 6: ... 7: System importiert den Titel.	
Alternative Ablaufschritte	<div data-bbox="564 335 1916 514"> <p>Offene Punkte:</p> <ul style="list-style-type: none"> - Name der Default-Playlist bei alternativem Ablaufschritt 3a? - Systemverhalten bei alternativem Ablaufschritt 7a? </div> <div data-bbox="434 628 1729 999"> <p>3a: Noch keine Playlist vorhanden → System erstellt automatisch eine Default-Playlist</p> <p>...</p> <p>4a: Titel bereits vorhanden → Korrektur durch Anwender oder Abbruch</p> <p>4b: ... andere Ausnahme ... → ...</p> <p>...</p> <p>7a: Import schlägt fehl → ???</p> <p>→ Sofort notieren</p> </div>	
Fragen	Name der Default-Playlist bei alternativem Ablaufschritt 3a?	
Fragen	Systemverhalten bei alternativem Ablaufschritt 7a?	
Kommentare		

→ Klären & nächste Iteration, ...

→ So viele Iterationen bis möglichst alles beschrieben

ANWENDUNGSFALLBESCHR. – WAS NOCH BEACHTEN?

- Vor- und Nachbedingungen:
 - Beschreibung als Zustände
 - Vorbed.: Benutzer ist eingeloggt,
System zeigt Hauptbildschirm
 - Nachbed.: System hat Auftrag geprüft und abgearbeitet
- Schritte:
 - Erster Schritt beginnt da wo die Vorbedingung aufgehört hat
→ 1. Benutzer wählt ... im Hauptbildschirm
 - Oft immer ein Wechsel zw. Benutzer* und System
→ 1. Benutzer wählt ... im Hauptbildschirm
2. System zeigt den Dialog ...
3. Benutzer wählt ...
4. System ...
- Immer nur EIN Anwendungsfall pro Schablone!

ID: Name	UC01: Titel hinzufügen
Ziel	Der Titel ist in der gewünschten Playlist enthalten.
Akteure	Anwender
Status	Entwurf
Verwendete UseCases	-
Auslöser	Kauf/Erhalt eines neuen Titels.
Vorbedingungen	Titel muss in Form einer Audio-Datei im Datei-System vorliegen. Das System ist geöffnet und zeigt das Hauptfenster
Nachbedingungen (Ergebnis)	Erfolg: Keine Veränderung an der Original-Audio-Datei, System enthält die Informationen über den Titel und die zugehörigen Audio-Daten. Misserfolg: Keine Veränderung am System.
Standardablauf	1: Anwender klickt auf die Funktion „Titel hinzufügen“. 2: Das System zeigt den Dialog „Titel hinzufügen“ (DIALOG_XY) 3: Anwender wählt eine Playlist aus. 4: ... 5: ... 6: ... 7: System importiert den Titel.
Alternative Ablaufschritte	3a: Noch keine Playlist vorhanden → System erstellt automatisch eine Default-Playlist ... 4a: Titel bereits vorhanden → Korrektur durch Anwender oder Abbruch 4b: ... andere Ausnahme ... → 7a: Import schlägt fehl → ??? ...
Zeitverhalten	Keine Einschränkung
Verfügbarkeit	Muss immer möglich sein
Fragen	
Kommentare	

*(oder anderem Akteur!)

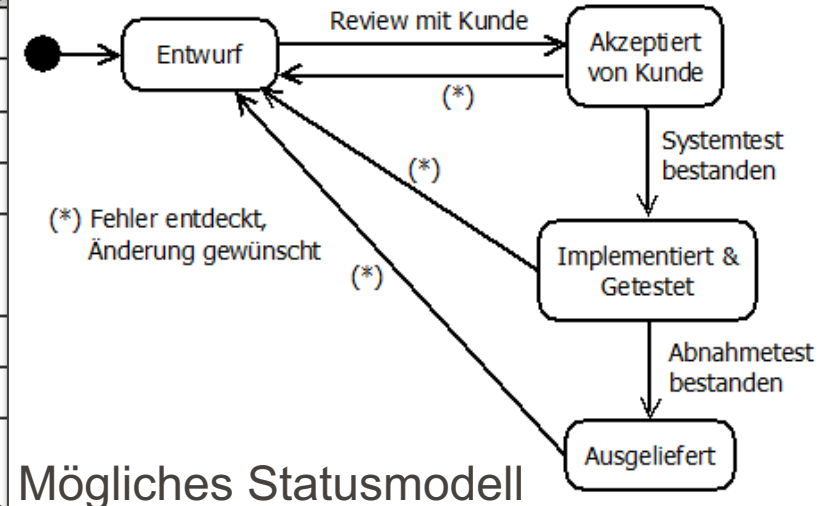
ANWENDUNGSFÄLLE – ANWENDUNGSFALLSCHABLONE



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

Mögliches Statusmodell:

ID: Name	
Ziel	
Akteure	
Status	
Verwendete UseCases	
Auslöser	
Vorbedingungen	
Nachbedingungen (Ergebnis)	Erfolg: Misserfolg:
Standardablauf	
Alternative Ablaufschritte	
Zeitverhalten	
Verfügbarkeit	
Fragen	
Kommentare	

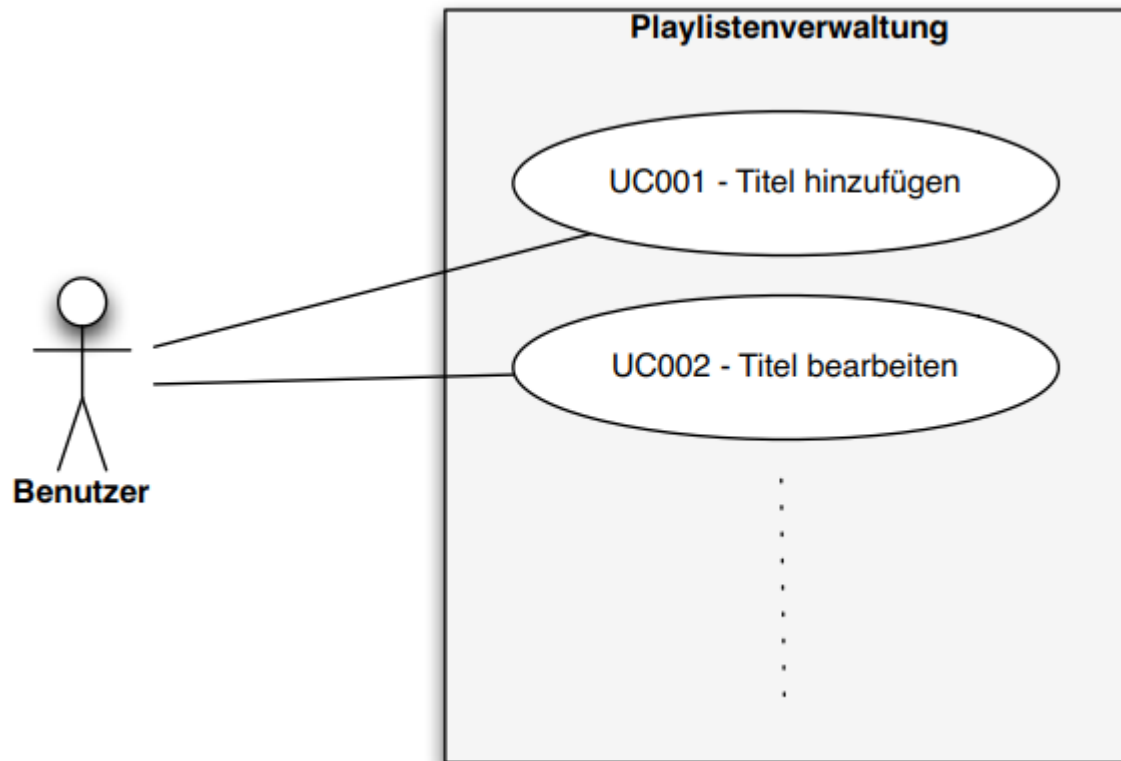


ANWENDUNGSFALLDIAGRAMM – BEISPIEL (MP3-PLAYER)



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

BSP: UseCase-Diagramm



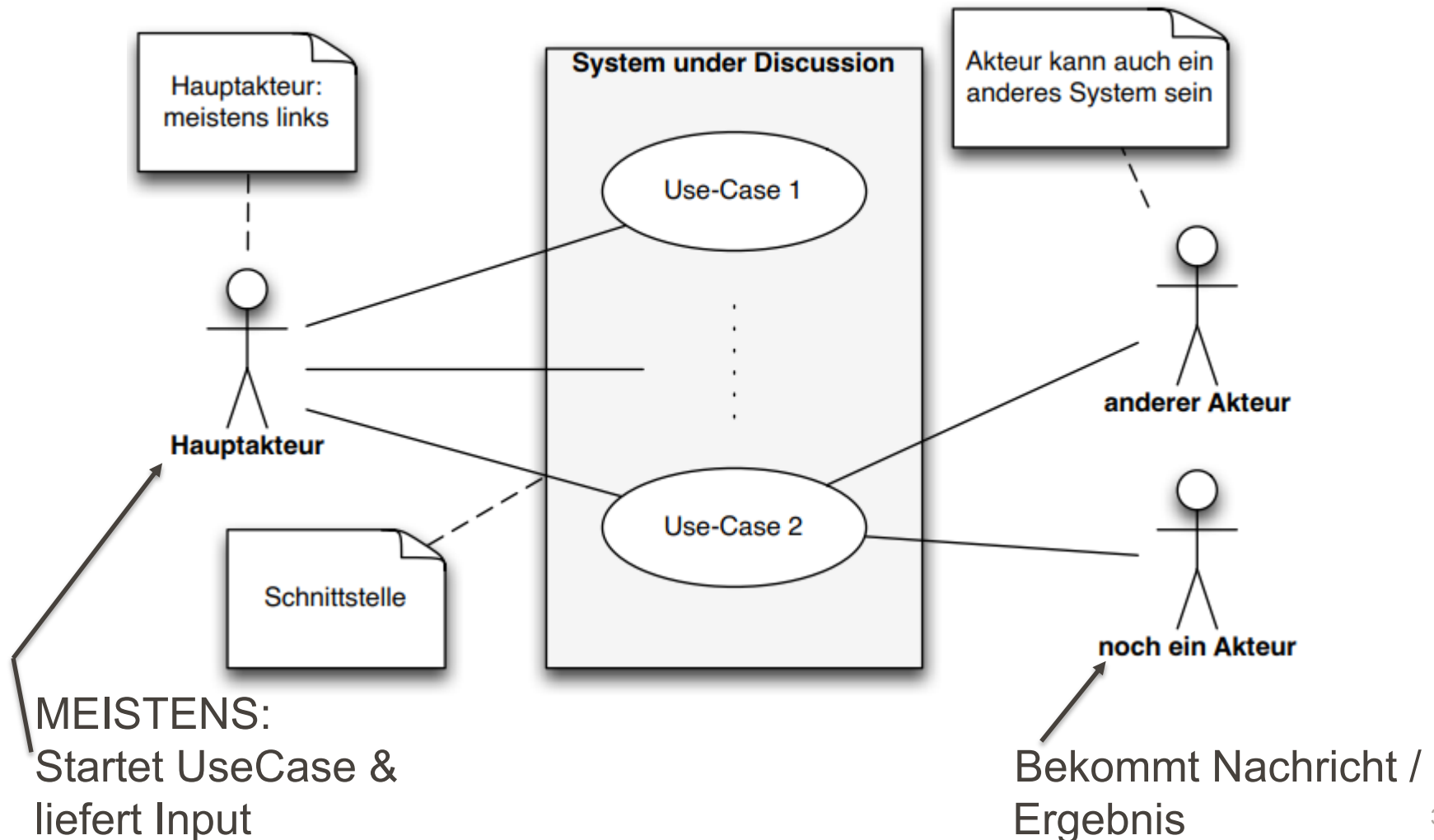
→ Auch Teil der UML zur Übersicht, über Anwendungsfälle

UML – ANWENDUNGSFALLDIAGRAMM



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

Allgemeine Form des UseCase-Diagramms:



UML – ANWENDUNGSFALLDIAGRAMM



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

- Das Anwendungsfalldiagramm (UseCase-Diagramm) enthält:
 - Auflistung der Namen der Anwendungsfälle
 - Auflistung der Akteure
 - Beteiligung von Akteuren an Anwendungsfällen
 - <<extends>>, <<includes>> (sparsam verwenden)

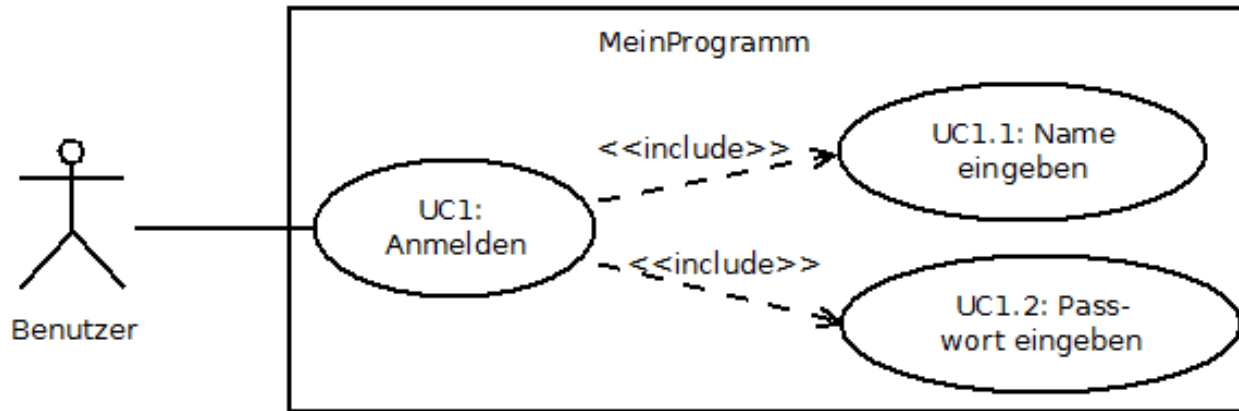
 Ein Anwendungsfalldiagramm ist keine:

- Darstellung von Ablaufschritten
- Darstellung zeitlicher Zusammenhänge
- Keine Reihenfolge!
- Ein Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!
- Ein Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!
- Ein Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!
- Ein Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!

→ BESCHREIBUNG EINES ANWENDUNGSFALLS:



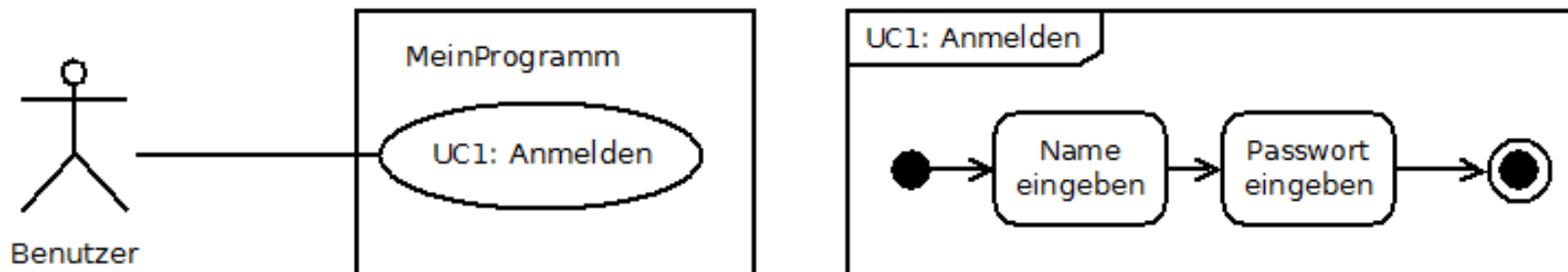
- Was halten Sie von diesem Use Case Diagramm?



⚠ Definiert nicht, dass UC1.1 vor UC1.2

⚠ Viel zu feingranular → Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!

→ Dann besser so:



→ BESCHREIBUNG EINES ANWENDUNGSFALLS:



Hochschule RheinMain

Univ	UC01: Titel hinzufügen
Wies	Ziel: Der Titel ist in der gewünschten Playlist enthalten.
	Aktore: Anwender
	Status: Entwurf
	Voraussetzungen: -
	Auslöser: Kauf/Einfuhr eines neuen Titels.
	UML-Klass: Titel muss in Form einer Audio-Datei im Daten-System vorliegen.
	Verbedingungen: Das System ist geöffnet und zeigt das Hauptfenster.
	Ergebnis: Erfolg: Keine Veränderung an der Original-Audio-Datei. System enthält die Informationen über den Titel und die zugehörigen Audio-Daten.
	Misserfolg: Keine Veränderung am System.
	Standardablauf: 1. Anwender klickt auf die Funktion „Titel hinzufügen“.
	2. Das System zeigt dem Dialog „Titel hinzufügen“ (DIALOG_XY).
	3. Anwender wählt eine Playlist aus.
	4: ...
	5: ...
	6: ...
	7. System importiert den Titel.
	Alternativ Ablaufschritte: 3a. Nach keine Playlist vorhanden → System erstellt automatisch eine Default-Playlist
	... 4a. Titel bereits vorhanden → Korrektur durch Anwender oder Abbruch
	4b: ... andere Ausnahme ... → ...
	7a. Import schlägt fehl → ???
	...
	Zeiterhalten: Keine Einschränkung
	Verfügbarkeit: Muss immer möglich sein
	Fragen: ...
	Kommentare: ...

- **Wichtigste Form: Textuelle Beschreibung** →
- Mögliche (zusätzliche) Formen:
 - Anwendungsfalldiagramm
 - 1 Anwendungsfall → 1 Aktivitätsdiagramm
 - 1 Anwendungsfall → 1 Zustandsdiagramm
 - mehrere Anwendungsfälle → 1 Aktivitätsdiagramm
 - mehrere Anwendungsfälle → 1 Zustandsdiagramm
 - Auch möglich: Sequenzdiagramm
 - BSP: Zustandsdiagramme für Sitzheizung, ...



Vorsicht:

- Ein Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!
- Ein Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!
- Ein Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!



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

04

Nichtfunktionale Anforderungen

Ziel:

Nichtfunktionale Anforderungen erfassen



NICHTFUNKTIONALE ANFORDERUNGEN



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

- Übliche Kategorien:
 - Performance & Zeitverhalten
 - Veränderbarkeit, Wartbarkeit
 - Bedienbarkeit
 - Sicherheit (Security & Safety)
 - Testbarkeit
 - Korrektheit
 - Verfügbarkeit, Zuverlässigkeit
 - ...
- Übliche Beschreibungsform: Textuelle Beschreibung
- Aber auch in Modellen
 - Z.B.: Zeitverhalten als Constraints in UML-Diagrammen

NICHTFUNKTIONALE ANFORDERUNGEN



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

 Vorsicht:

- Werden sehr gerne vernachlässigt!
 - Leider auch hier etwas aus Zeitgründen!
- Viele Projekte scheitern, weil wichtige NFkt. Anforderungen vernachlässigt wurden
 - Z.B.: SW zu langsam (Performance), SW schlecht bedienbar, SW nicht mehr wartbar, SW nicht mehr änderbar
 - Oft Kombination aus mehreren
- Problem: Oft schwer greifbar und werden deshalb vergessen

→ Mehr dazu:

- Wahlpflichtveranstaltung Anforderungsmanagement (im WS)



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

05 GUI

Ziel:
Kundenanforderungen für die GUI erfassen



GUI IN DER ANFORDERUNGSANALYSE



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

- Was sollte man zur GUI in der Anforderungsanalyse aufschreiben?
 - Zumindest die Informationen, die benötigt werden, um
 - Anwendungsfälle
 - Fachmodellzu verstehen.
 - Oft: Vorgriff auf Entwurf
 - Grober Entwurf der GUI
 - Fenster mit UI-Elementen (Werkzeug: z.B. UI-Editor)
 - (grobe) Navigation (z.B. mit Zustandsdiagramm)
 - Klick-Prototyp (z.B. mit HTML)

GUI IN DER ANFORDERUNGSANALYSE



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

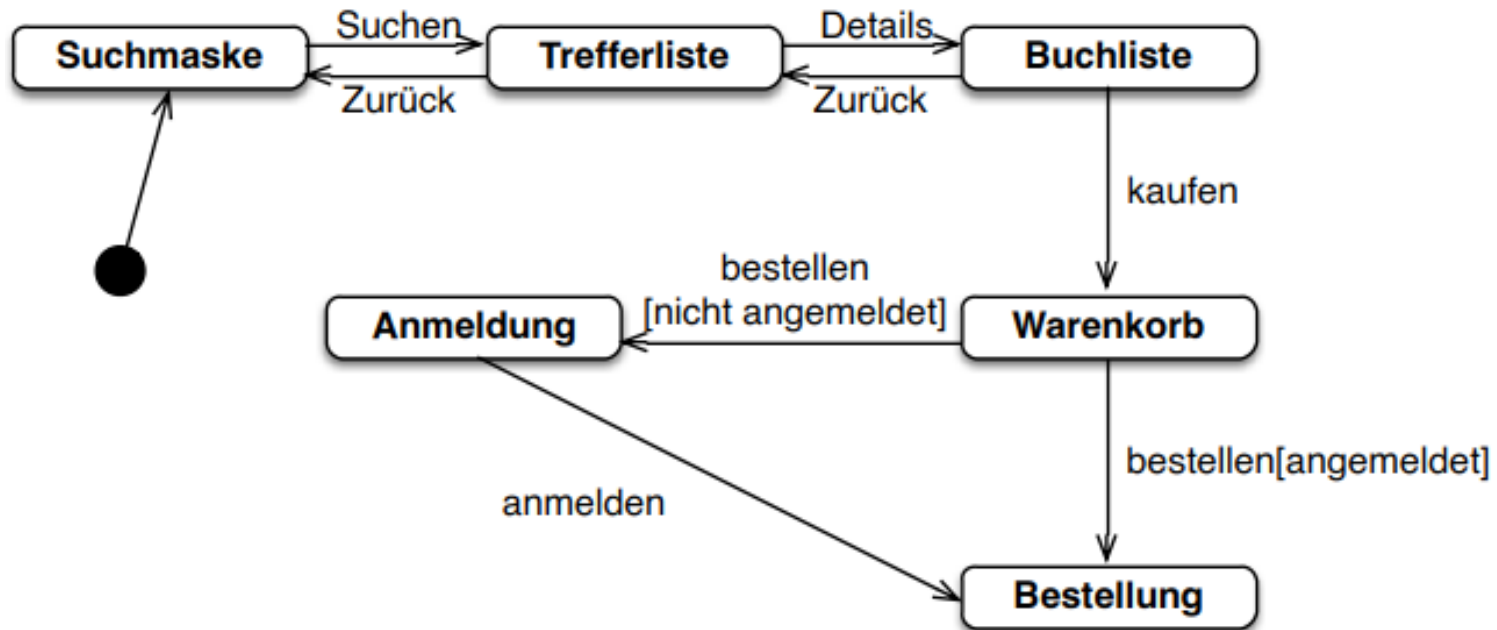
- Warum sollte man bereits bei der Anforderungsanalyse auf die GUI eingehen?
 - Risiken minimieren
- Schlechte Bedienbarkeit führt oft zum Scheitern von Projekten
 - BSP: IBM Lotus Notes
 - Super Technologiekonzept → Perfekt für jede IT-Abteilung
 - ABER: Bedienbarkeit fürchterlich
 - Z.B.: F3 → Beenden des Programms ohne Speichern
- Gute Bedienbarkeit kann über vieles hinwegtrösten → Hype
 - BSP: Apple
 - Super Bedienbarkeit
 - Völlig überteuerte Hardware, Kaum Reparaturmöglichkeiten, Inkompatibilitäten / Vendor-Lockin, Teurere Onlineshoppingpreise, ...

GUI IN DER ANFORDERUNGSANALYSE



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

- Beispiel: Navigationsregeln für Online-Buchladen



→ Zustandsdiagramm:

- Zustände \triangleq Bildschirmmasken / Bedienabschnitten in Bildschirmmasken
- Übergänge \triangleq Aktionen (z.B. Klicks auf Buttons, ...)

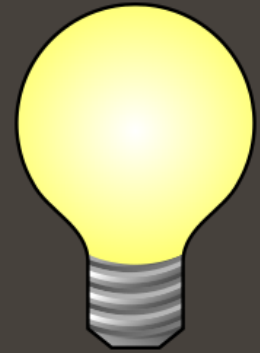


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

06

Qualitätssicherung

Ziel:
Vorgriff auf spätere Vorlesung
→ Jedoch fängt hier schon alles an





- Wie können wir sicherstellen, dass . . .
 - die Anforderungsanalyse richtig durchgeführt wurde?
 - die (noch zu bauende) Anwendung die Anforderungen erfüllt?
- Antworten:
 - Überprüfung der erstellten Dokumente → Reviews
 - Vorbereitung des funktionalen Systemtests
 - Input: Use Cases, Fachmodell, GUI-Entwurf
 - Output: Testfälle + Testskripte + ggfs. Testwerkzeuge
 - Vorbereitung der nichtfunktionalen Systemtests
 - Input: Nichtfunktionale Anforderungen
 - Output: Testfälle + Testskripte + ggfs. Testwerkzeuge

→ Details dazu später bei der Vorlesung über Testen



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

07

Das war's noch lange nicht

Ziel:

Ausblick auf Wahlpflichtvorlesung Anforderungsmanagement



ANFORDERUNGSANALYSE – DAS WAR SEHR, SEHR KURZ!



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

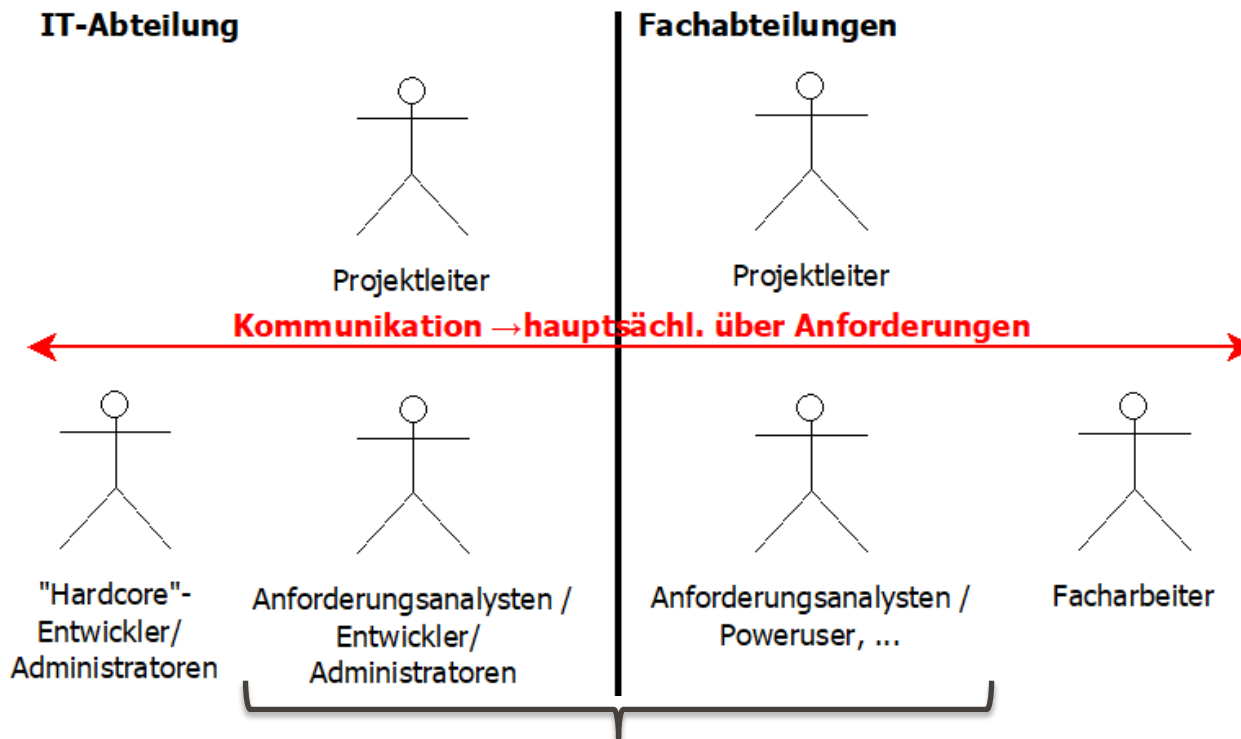
- Das war alles sehr, sehr kurz
 - ABER: Thema ist sehr wichtig, weil Anforderungen die Basis aller weiteren Schritte bilden
 - Gerade auch Kernaufgabengebiet der WI
 - Siehe folgende Folie
- In der Wahlpflichtvorlesung Anforderungsmanagement können Sie wesentlich mehr dazu erfahren
- Wie finde ich Anforderungen überhaupt?
 - Muss man aufdecken
 - Sehr viel Psychologie & Kommunikation
 - Wie schreibe ich Anforderungen möglichst eindeutig auf?
 - Nichtfunktionale Anforderungen
 - Wie manage ich Anforderungen und Anforderungsänderungen?
 - ...

ANFORDERUNGSMANAGEMENT – EIN KERNGEBIET DER WI



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

- Kernaufgabengebiet der WI
 - IDEE der WI: Übersetzer Fachabteilung ↔ IT-Abteilung:



Typisches Berufsbild WI

→ Sie können natürlich auch etwas anderes machen
(Gerade unser technischer Fokus ist sehr gefragt)



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

08

Fazit

Ziel:
Was haben wir damit gewonnen?





WAS HABEN WIR GELERNT?

- Anforderungen sind sehr wichtig
 - Müssen ja wissen was wir eigentlich entwickeln wollen
- Verschiedene Aspekte
 - Fachmodell
 - Anwendungsfälle
 - Nichtfunktionale Anforderungen
 - GUI
 - Qualitätssicherung
- Leider haben wir viel zu wenig Zeit
 - Das Thema ist viel wichtiger
 - Wird auch so gerne unterschätzt

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].
- Larman, C.: Applying UML and Patterns [30 BF 500 78].



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

AUF GEHT'S!!

SELBER MACHEN UND LERNEN!!

