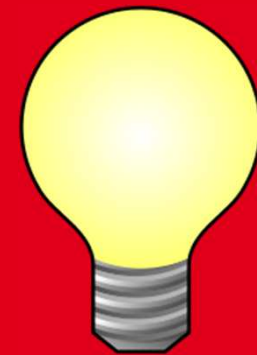


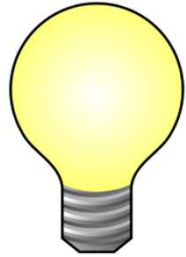


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

04.12.2018 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

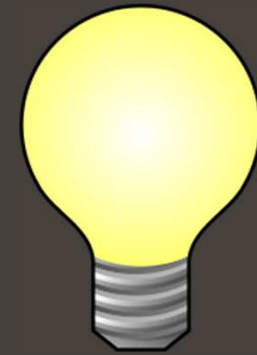
Fazit



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

01 EINFÜHRUNG INS THEMA

Ziel:
Die Eckpunkte des Themas kennenlernen

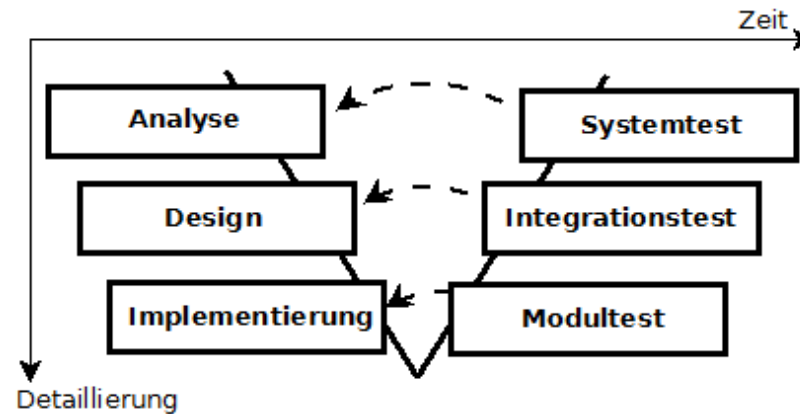


WORUM GEHT'S?



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

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



- Heute:
 - Anforderungsanalyse

ANFORDERUNGSANALYSE – EINFÜHRUNG



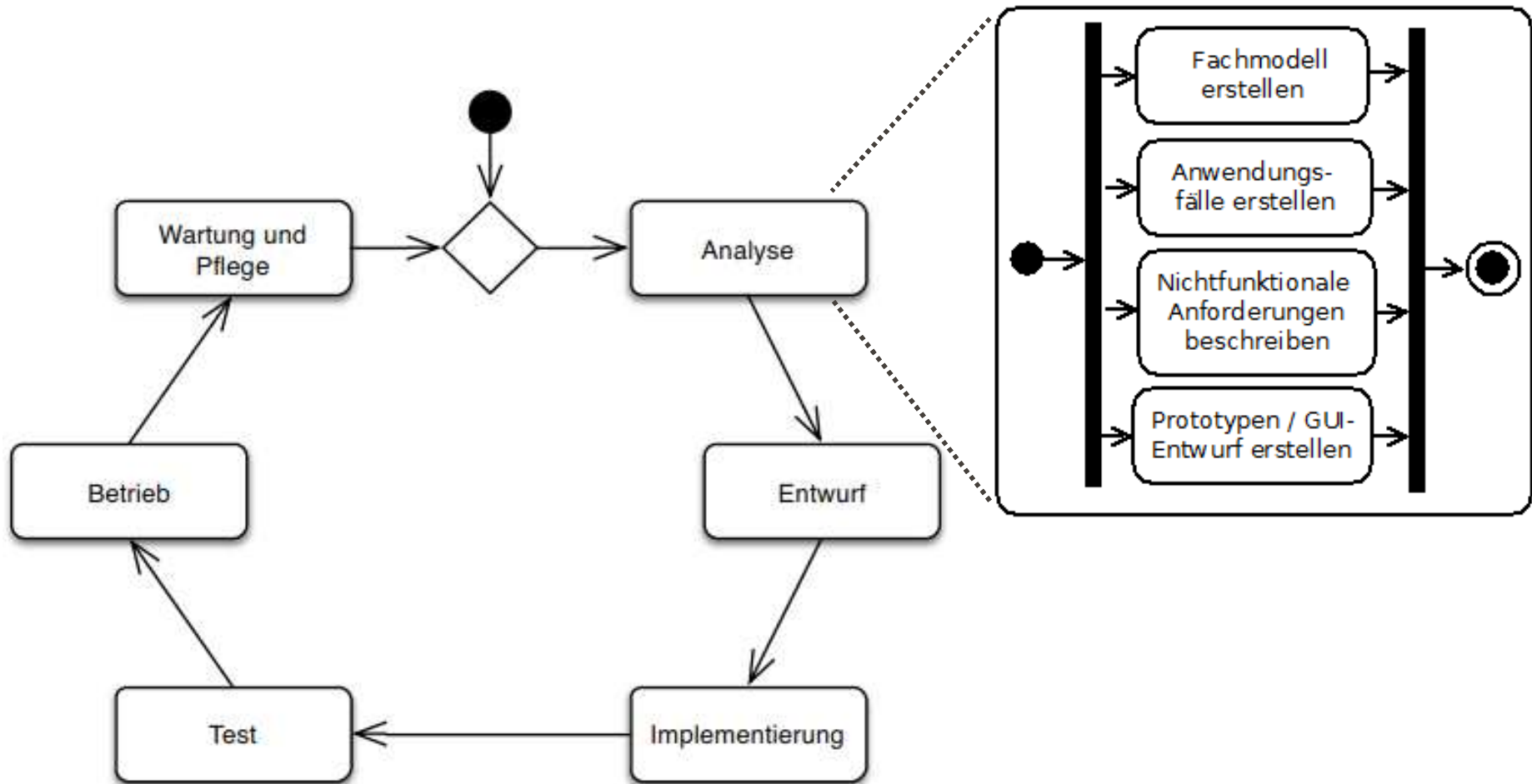
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- 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
 - nicht-funktionale Anforderungen (z.B. Performance, Sicherheit,...)
 - ggfs. Prototypen, GUI-Entwurf, . . .

ANFORDERUNGSANALYSE – UNTERAKTIVITÄTEN:



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



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)
 - ABER:
 - Sehr einfaches Beispiel
 - Sehr eingängig (jeder von Ihnen kennt das)



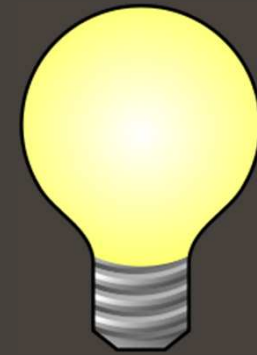
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

02

Das Fachmodell

Ziel:

Das Fachmodell kennenlernen

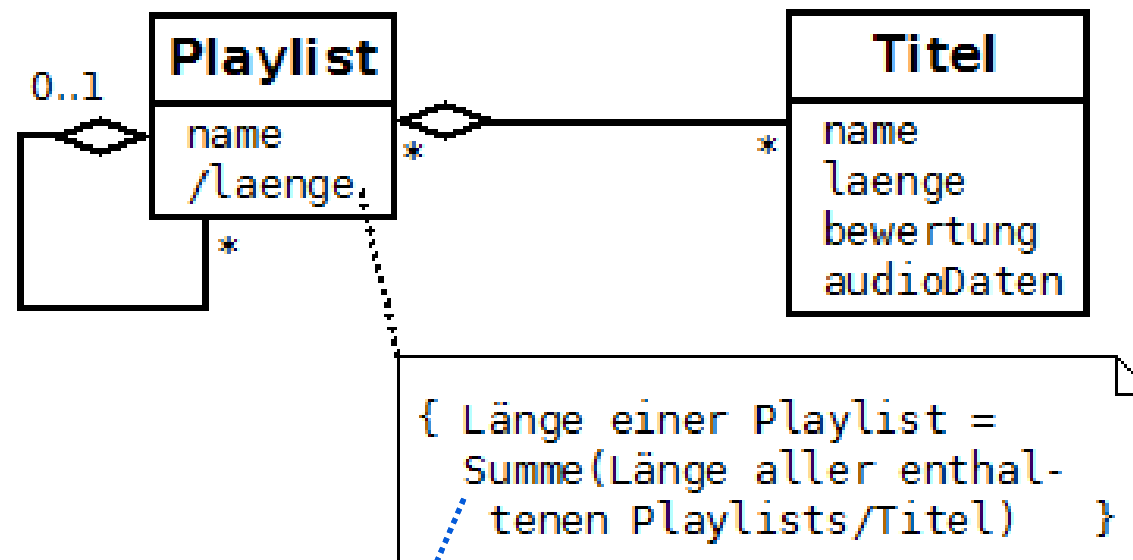


FACHMODELL – BEISPIEL



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?

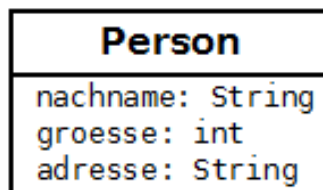


Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

⚠ Vorsicht

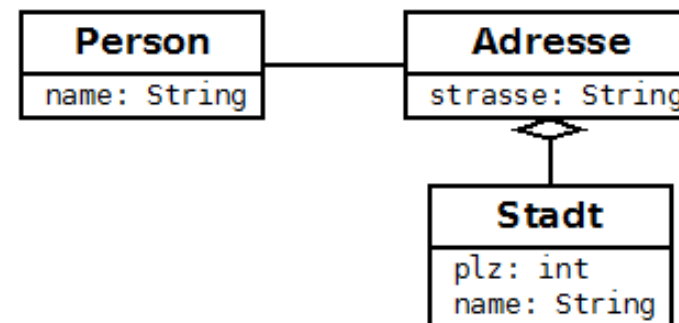
- i.d.R.: Klassen-Diagramm für Fachmodell \neq Klassen-Diagramm für Java-Klassen
- gleiche Notation für völlig verschiedene Anwendungsbereiche!
- Fachmodell bildet nur die fachlichen Zusammenhänge ab
- Im Design kommen noch viele andere Aspekte hinzu
 - Fachmodell bildet aber meist die Ausgangsbasis

Fachmodell:



↔

“Fachmodell” im Design:



→ In Zukunft Domänenmodell
genannt

(vgl. auch Vorlesung zu Objektdiagramm)

FACHMODELL – WIE KOMMT MAN DAZU?



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

- Bewährte Vorgehensweisen
 - Bewährtes aus der Vorlesung „Datenbanken“
 - z.B.: Normalformen
 - Analysemuster
- 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
 - 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
 - irrelevante Begriffe streichen
 - vage Begriffe streichen oder präzisieren
4. Klassen mit Attributen und Beziehungen (Assoziationen) erstellen

* R. J. Abbott: „Program Design by Informal English Descriptions“. Comm. of the ACM, 26, #11, 1983. 13

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



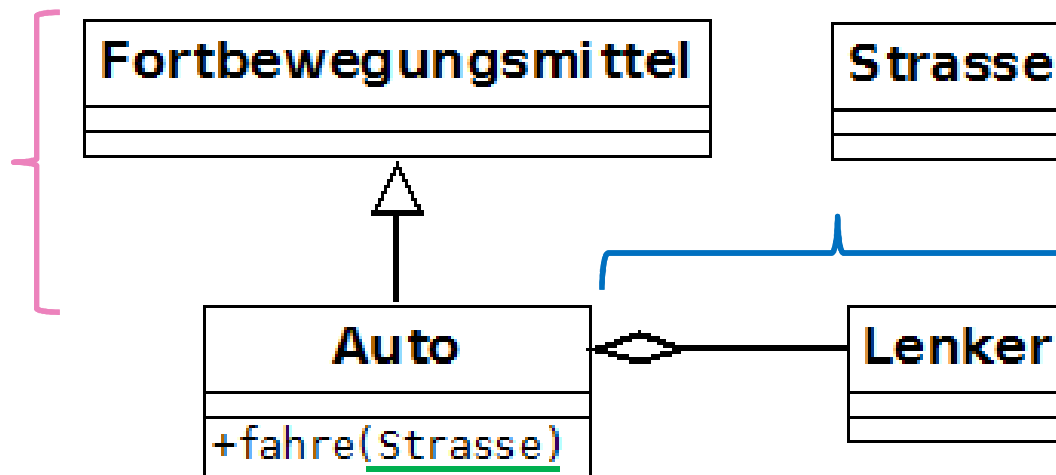
Deutet auf Attribut hin



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.

ANSÄTZE AM BESTEN KOMBINIEREN



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

Mögliche Kombination der Ansätze:

1. Textanalyse nach Abott
 2. Beziehungen nach B. Stroustrup
 3. Im Zweifelsfalls / wenn es nicht klar ist?
→ Objektdiagramme verschiedener Situationen zeichnen
- + Evtl. andere Analysemuster
(siehe spätere Vorlesung zu Mustern)



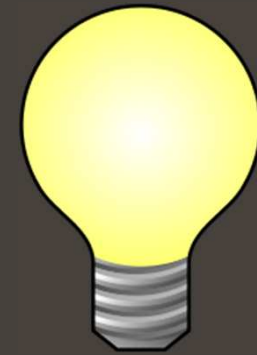
Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

03

Anwendungsfälle (Use Cases)

Ziel:

In Anwendungsfällen spezifizieren lernen



ANWENDUNGSFÄLLE (USE CASES)



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

- Definition Anwendungsfälle (Use Cases):
 - Beschreibung der dynamischen Aspekte (Daten**verarbeitung**) 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 einen,
 - 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 + Unique ID
- Ziel
- Beteiligte Akteure
- Verwendete (andere) Anwendungsfälle
- Auslöser, Vorbedingung, Nachbedingung (im Erfolgs-Fall und im Misserfolgs-Fall)
- Standardablauf (Szenario, in dem alles gut und normal verläuft)
- Alternative Ablaufschritte
 - Auflistung **aller** Ausnahmen/Fehlerfälle
 - Verhalten in Ausnahmesituation
- Zeitverhalten, Verfügbarkeit
- Fragen & Kommentare

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	

ANWENDUNGSFALL – BEISPIEL



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

BSP:

ID: Name	UC01: 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.
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.

ANWENDUNGSFALL – BEISPIEL



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

BSP:

Standardablauf	1: Anwender öffnet den Dialog für die Funktion „Titel hinzufügen“. 2: Anwender wählt eine Playlist aus. 3: ... 4: ... 5: ... 6: System importiert den Titel.
Alternative Ablaufschritte	2a: noch keine Playlist vorhanden ... 4a: Titel bereits vorhanden 4b: ... andere Ausnahme 6a: Import schlägt fehl ...
Zeitverhalten	Keine Einschränkung
Verfügbarkeit	Muss immer möglich sein
Fragen	
Kommentare	

→ 1. Iteration

ANWENDUNGSFALL – BEISPIEL



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

BSP:

Standardablauf	1: Anwender öffnet den Dialog für die Funktion „Titel hinzufügen“. 2: Anwender wählt eine Playlist aus. 3: ... 4: ... 5: ... 6: System importiert den Titel.
Alternative Ablaufschritte	2a: noch keine Playlist vorhanden ... 4a: Titel bereits vorhanden
Alternative Ablaufschritte	2a: Noch keine Playlist vorhanden → System erstellt automatisch eine Default-Playlist ... 4a: Titel bereits vorhanden → Korrektur durch Anwender oder Abbruch 4b: ... andere Ausnahme ... → 6a: Import schlägt fehl → ??? ...

Das ist noch sehr ungenau
→ Reaktionen fehlen

→ 2. Iteration

ANWENDUNGSFALL – BEISPIEL



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

BSP:

Standardablauf	<div data-bbox="728 432 2074 608"> Offene Punkte: - Name der Default-Playlist bei alternativem Ablaufschritt 2a? - Systemverhalten bei alternativem Ablaufschritt 6a? </div> 1: A 2: A 3: . 4: ... 5: ... 6: System importiert den Titel.
Alternative Ablaufschritte	2a: Noch keine Playlist vorhanden → System erstellt automatisch eine Default-Playlist ... 4a: Titel bereits vorhanden → Korrektur durch Anwender oder Abbruch 4b: ... andere Ausnahme ... → 6a: Import schlägt fehl → ??? <div data-bbox="1377 954 1881 1013">→ Sofort notieren</div>
Fragen	Name der Default-Playlist bei alternativem Ablaufschritt 2a? Systemverhalten bei alternativem Ablaufschritt 6a?
Fragen	
Kommentare	

→ Klären & nächste Iteration, ...

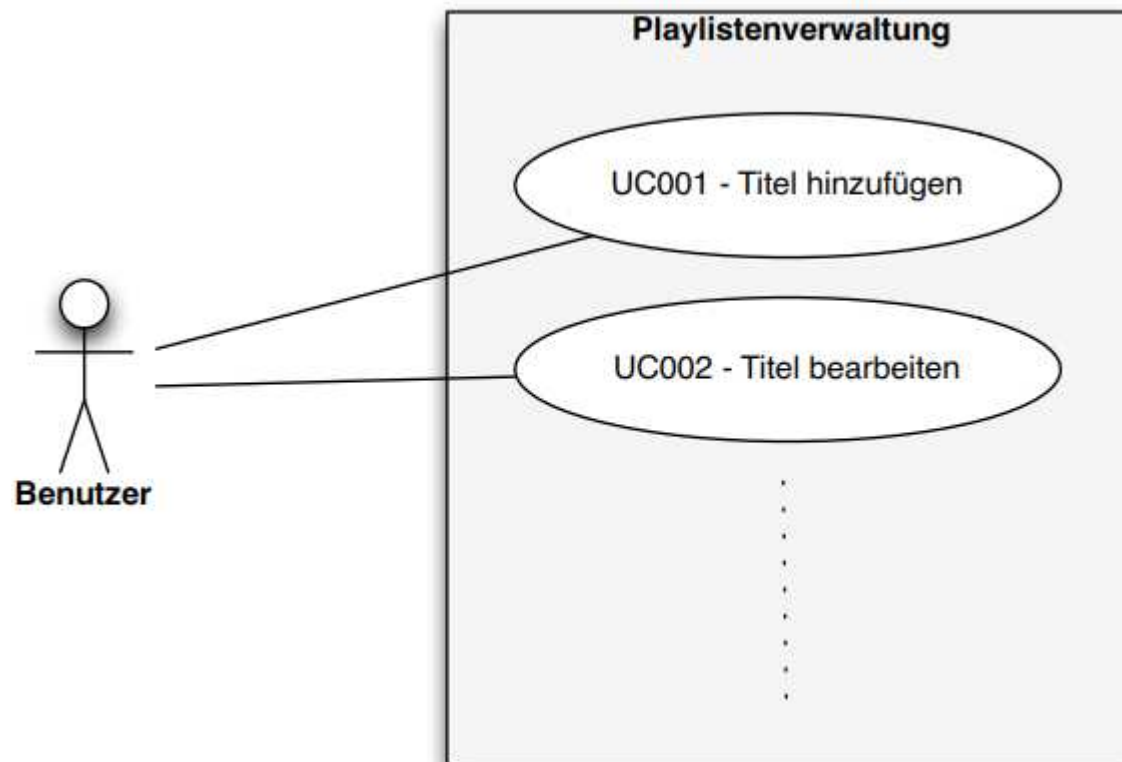
→ So viele Iterationen bis möglichst alles beschrieben

ANWENDUNGSFALL – BEISPIEL



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

BSP: UseCase-Diagramm



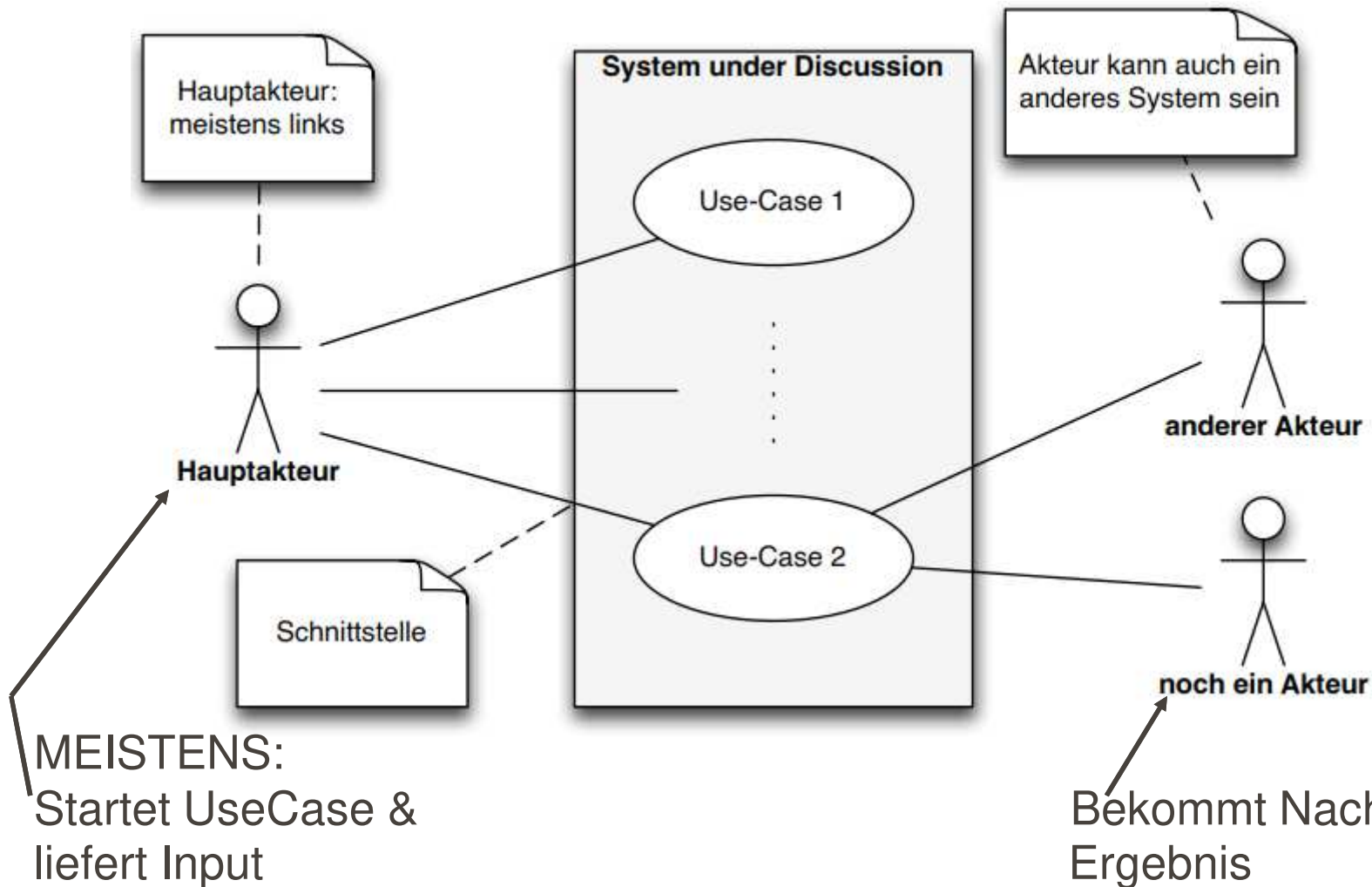
→ Auch Teil der UML zur Übersicht, über Anwendungsfälle

ANWENDUNGSFALLDIAGRAMM



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

Allgemeine Form des UseCase-Diagramms:



ANWENDUNGSFALLDIAGRAMM



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

- Das Anwendungsfalldiagramm (UseCase-Diagram) 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:



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

- **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:

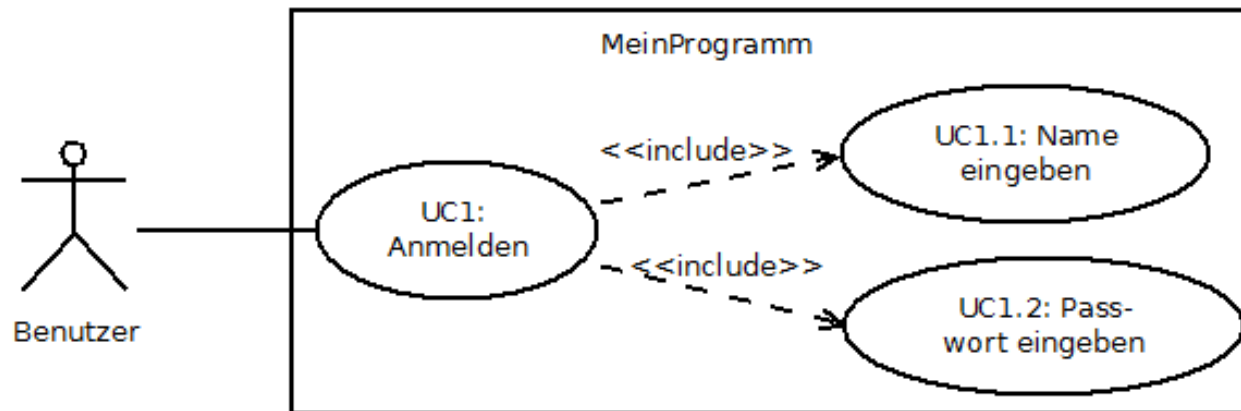
- Ein Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!
- Ein Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!
- Ein Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!

→ BESCHREIBUNG EINES ANWENDUNGSFALLS:



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

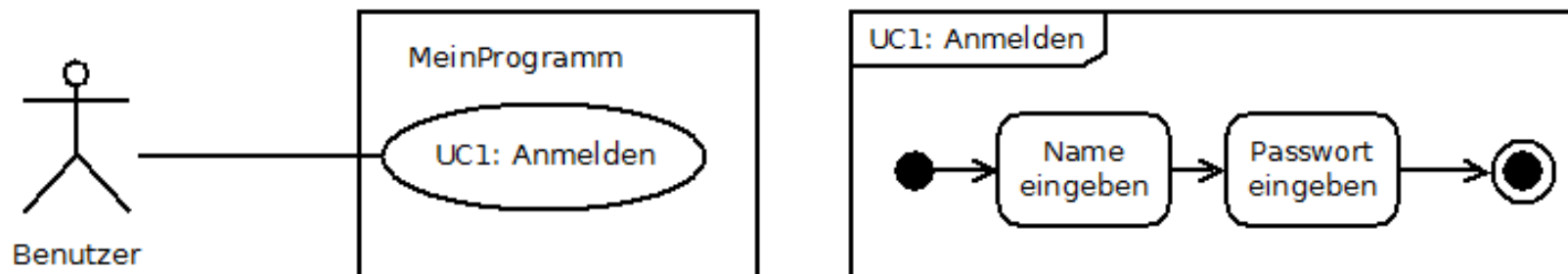
- Was halten Sie von diesem Use Case Diagramm?



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

⚠ Viel zu feingranular → Anwendungsfalldiagramm ist kein Aktivitätsdiagramm!

→ Dann besser so:





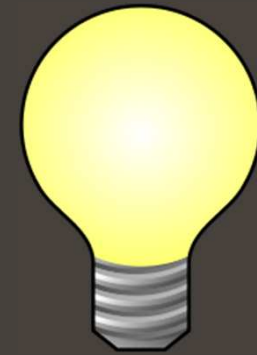
Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

04

Nichtfunktionale Anforderungen

Ziel:

Nichtfunktionale Anforderungen erfassen



NICHTFUNKTIONALE ANFORDERUNGEN



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

- Übliche Kategorien:
 - Performance & Zeitverhalten
 - Veränderbarkeit
 - 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:

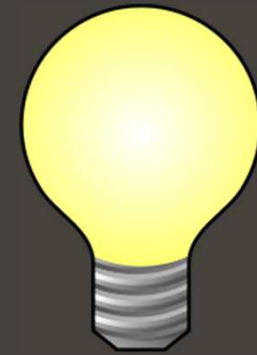
- Wahlpflichtveranstaltung Anforderungsmanagement im Sommersemester



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 Analyse & 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

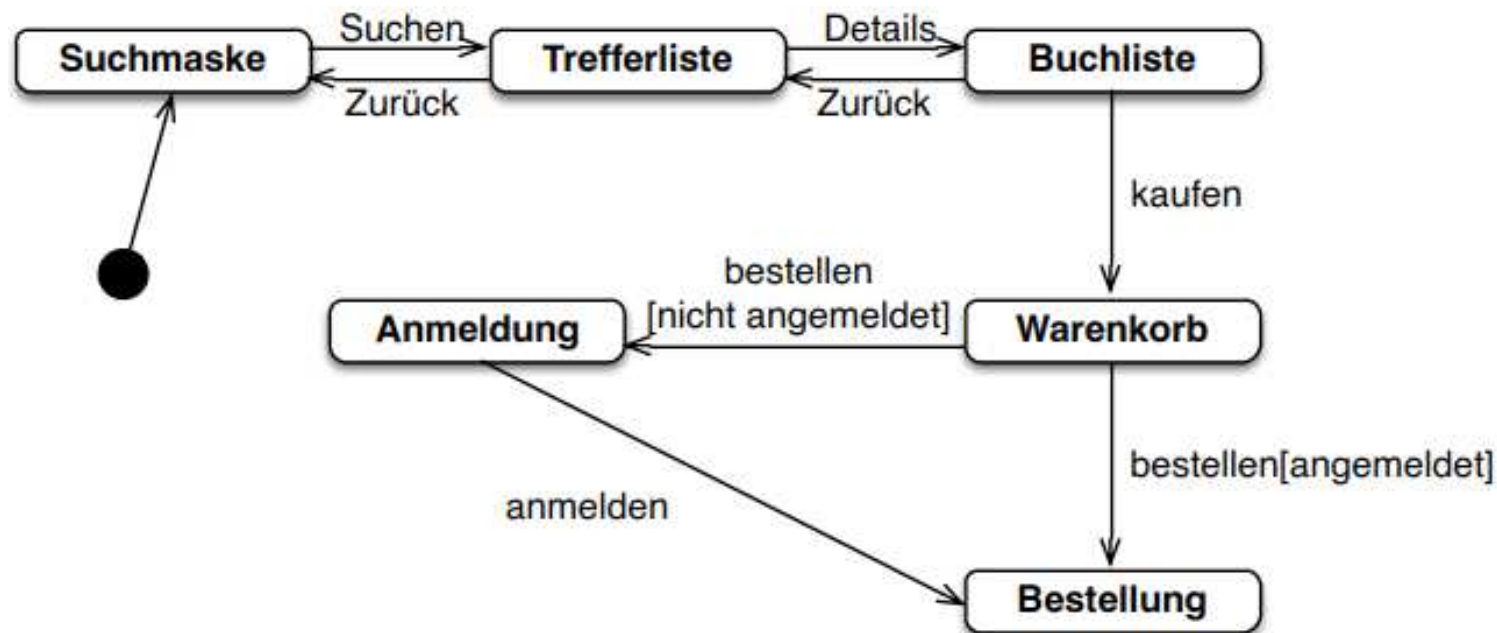
- Warum sollte man bereits bei der Anforderungsanalyse auf die GUI eingehen?
 - Risiken minimieren
- Schlechte Bedienbarkeit führt oft zum Scheitern von Projekten
 - BSP: IBM Lotus Notes
 - Super Technologiekonzept → Perfekt für jede IT-Abteilung
 - Bedienbarkeit fürchterlich
 - Z.B.: F3 → Beenden des 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. Button-klicks)



Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

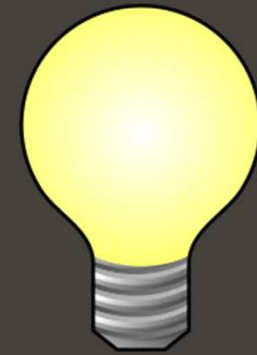
06

Qualitätssicherung

Ziel:

Vorgriff auf spätere Vorlesung

→ Jedoch fängt hier schon alles an



QUALITÄTSSICHERUNG



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

- 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 nicht-funktionalen Systemtests
 - Input: nicht-funktionale Anforderungen
 - Output: Testfälle + Testskripte + ggfs. Testwerkzeuge

→ Details dazu später bei der Vorlesung über Testen



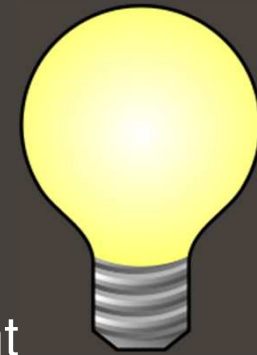
Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

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

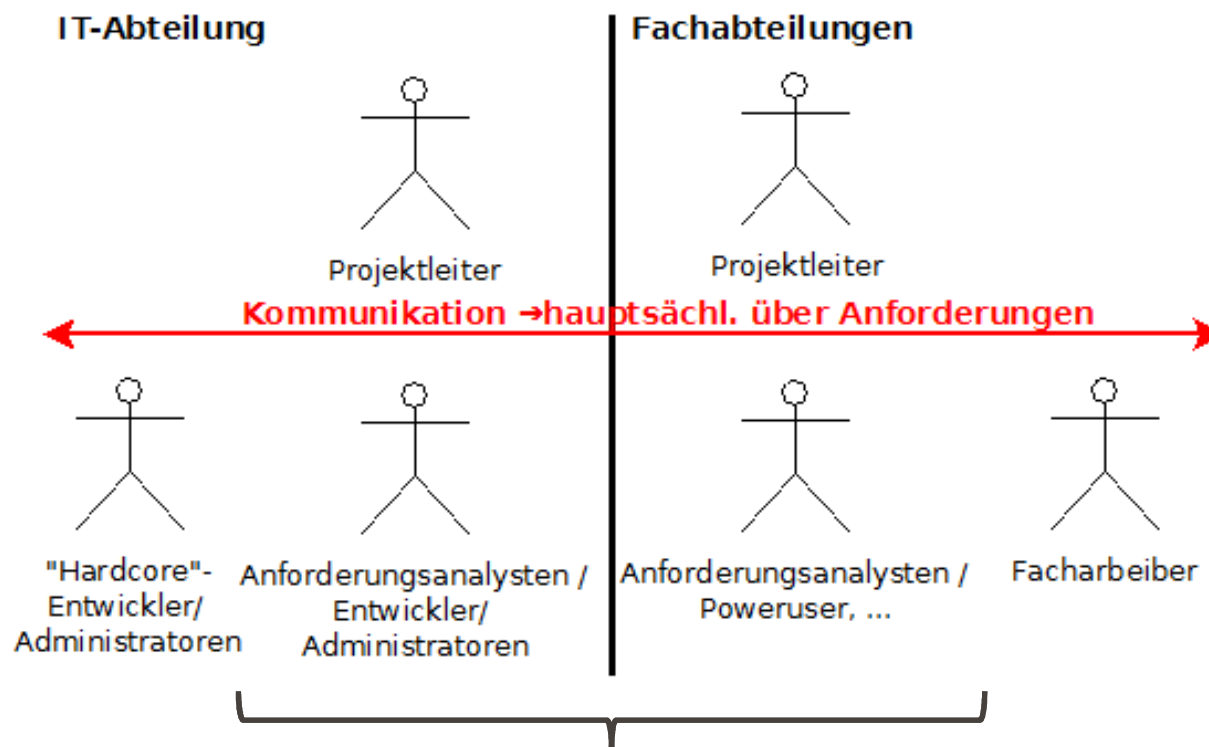
- Das war alles sehr, sehr kurz
 - Thema ist sehr wichtig, weil Anforderungen die Basis aller weiteren Schritte bilden
 - Gerade auch Kernaufgabengebiet der WI
 - Siehe folgende Folie
- In der Wahlpflichtvorlesung Anforderungsmanagement können Sie wesentlich mehr dazu erfahren
- Wie finde ich Anforderungen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

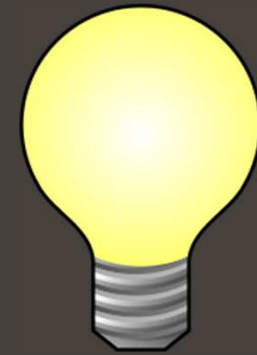
→ Sie können natürlich auch etwas anderes machen
(Gerade unser technischer Fokus ist sehr gefragt)



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].
- 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!!

