

SE Besprechung

Übung 3 – Softwareprozesse

SE, 20.10.09



Dustin Wüest

Aufgabe 1 – Systemkomponenten

- Ziel: Das System (das Problem) in mehrere, kleine Teillösungen zerlegen
- Komponenten, die (+/-) unabhängig voneinander entwickelt werden können
- Abhängigkeiten zwischen Komponenten zeigen (hilft, Verantwortlichkeiten festzulegen)

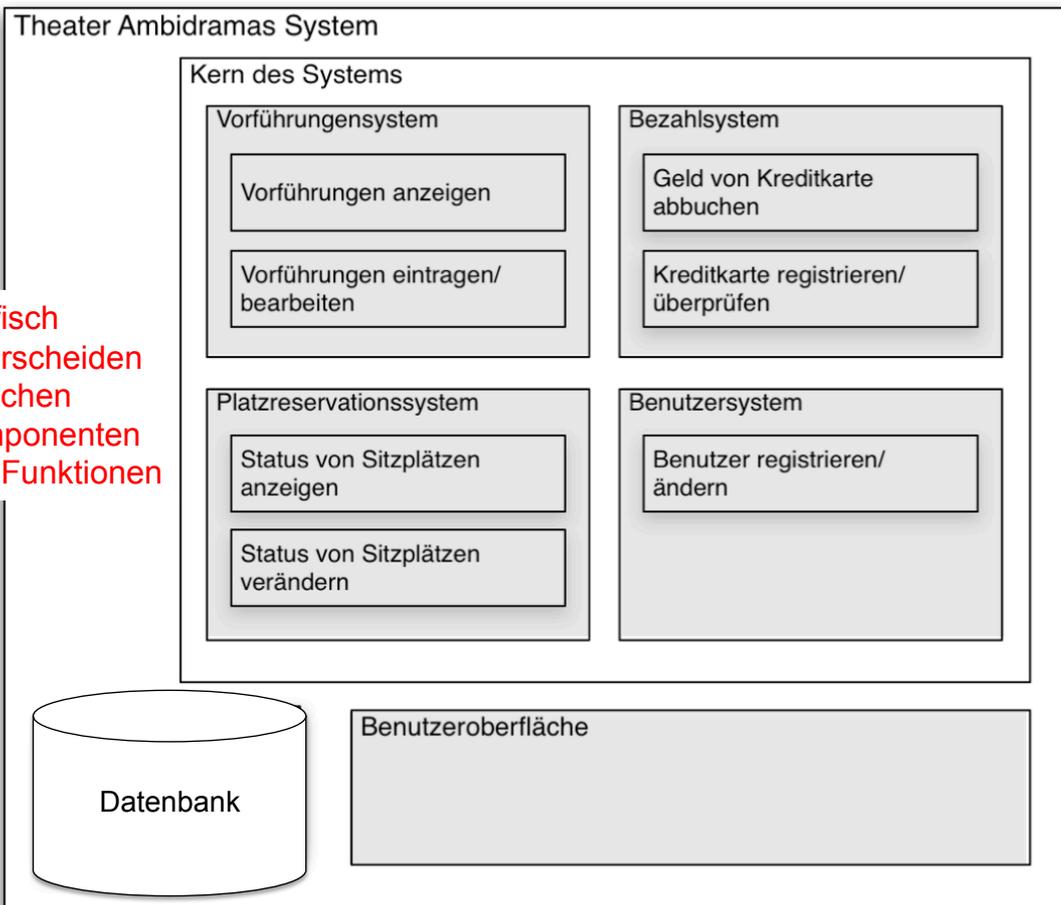
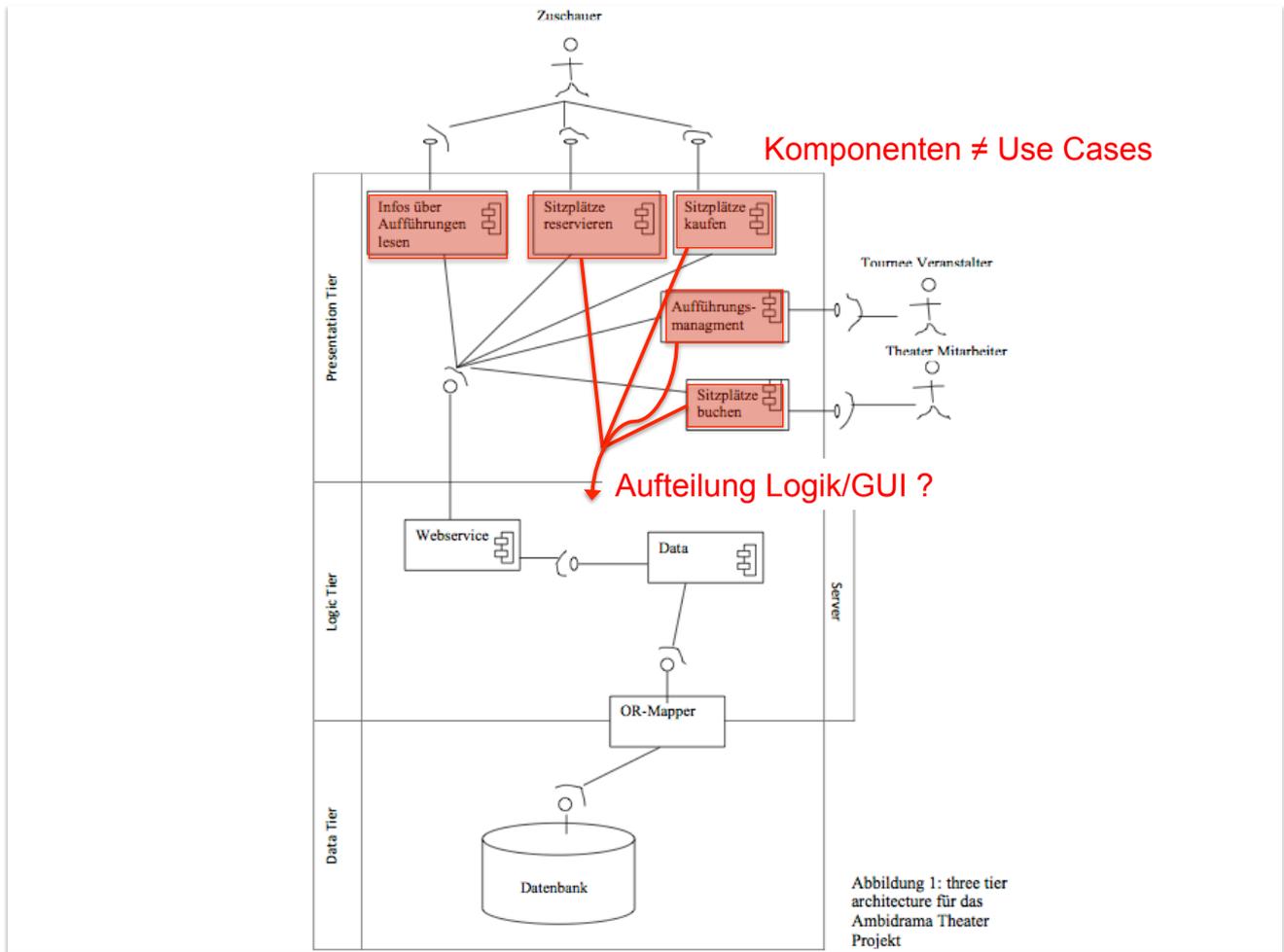
Aufgabe 1 – Systemkomponenten

- Hierarchische Zerlegung möglich, eine Komponente kann Sub-Komponenten besitzen
- GUI, Logik, DB Komponenten
- Beispiele für Komponenten:
Logik: Onlinekauf, Onlinereservation, Zahlungssystem, Adminkomponente, AufführungsMngtSystem, Rechteverwaltung, (Altes System), News-Dienst
Präsentation: GUI (ev. versch. GUIs für versch. Teilsysteme)
Daten: DB (Sitzplätze, Kundeninfos, Veranstaltungen)

Aufgabe 1 – Systemkomponenten

Kritik

- Einige Lösungen sehr kurz
- Namen der Komponenten und Funktionalität
- Komponenten \neq Use Cases
(strukturelle Teile des Systems \neq Szenarios aus User-Sicht)
- Hierarchische Dekomposition + Abhängigkeiten:
Diagramm zur Darstellung besonders geeignet (mit Legende)



Aufgabe 2 – Prozessmodelle

- Teil A: 3 Punkte
- Vor-/Nachteile bezüglich des Ambidramas Projekts
(z.B. Wachstumsmodell: ein Vorteil ist schrittweise Einführbarkeit. Ist das Ambidramas Projekt dafür geeignet, oder fällt dieser Vorteil nicht ins Gewicht?)
- Einfluss der Modellwahl auf die Projektlänge und Entwicklungskosten meiner Meinung nach z.T. überbewertet
- Kosten/Zeitaufwand bei Agiler SE gegenüber anderen Modellen ↑ oder ↓?

Aufgabe 2 – Prozessmodelle

- Teil B: 2 Punkte
Teil C: 2 Punkte
- Neben Modellwahl ist v.a. auch Begründung / Argumentation wichtig
- Klar / Verständlich schreiben

Aufgabe 2 – Prozessmodelle

- Kriterien für Modellwahl
 - Anforderungen klar / genau definierbar?
 - Werden sich Anforderungen mit der Zeit ändern?
 - Projektgrösse
 - Wichtigkeit (Sicherheitskritisches System?)
 - Entwicklungsrisiko
 - Know How Entwickler / Kunde
 - Verfügbarkeit und Kompetenz des Kunden
 - ...

Aufgabe 2 – Prozessmodelle

- **Phasenmodell**

System wird zuerst konzipiert, dann realisiert... (Projekt Lebenszyklus)

Benötigt:

Definierbare Anforderungen (egal ob bereits definiert oder nicht)

Kleines Entwicklungsrisiko (≠ Wichtigkeit)

Vorteil:

Gut für Projekt Management

Nachteil:

Schlechte Adäquatheit (Risiko)

Technische Risiken

Aufgabe 2 – Prozessmodelle

- **Wachstumsmodell**

Ein System wächst... (Systemverhalten)

Benötigt:

Einführbar in mehreren Schritten (Lieferung auch intern möglich)

Vorteil:

Sehr gut für Projekt Management

Früher Start (frühes Feedback von Usern, Motivation der Stakeholder)

Nachteil:

Gefahr, dass Software schlecht strukturiert ist

Aufgabe 2 – Prozessmodelle

- **Agile Softwareentwicklung**

Benötigt:

Entscheidungsbefugter, kompetenter Kundenvertreter vor Ort

Erfahrener Softwarearchitekt

Intensive Qualitätssicherung an der Quelle

Vorteil:

Beteiligung des Kunden

Nachteil:

Gefahr des Absturzes in die Ad-hoc Programmierung

Gefahr, dass Software schlecht strukturiert ist

Aufgabe 2 – Prozessmodelle

- Anmerkungen (1):

Nach der Entwicklung kann ein System erweitert werden, unabhängig vom gewählten Prozessmodell

(Das Phasenmodell hat Probleme mit Anforderungsänderungen **während** der Entwicklung)

Das Ziel aller Prozessmodelle ist es, Fehler früh zu finden und das Risiko zu minimieren (abhängig vom Modell kämpfen wir mit anderen Typen von Fehlern und Risiken)

(Phasenmodell → Spezifikationsfehler, Wachstumsmodell → Adäquatheitsfehler)

Aufgabe 2 – Prozessmodelle

- Anmerkungen (2):

Andere haben schon Ähnliches gemacht

= technisches Risiko gering

≠ Know How ist vorhanden

Wachstumsmodell ermöglicht auch Zeitplanung, Meilensteine und deren Überprüfung

Aufgabe 2 – Prozessmodelle

- Modellwahl



Aufgabe 2 – Prozessmodelle

- Ambidramas

18 Phasenmodell

5 Agile SE

3 Wachstumsmodell



Aufgabe 2 – Prozessmodelle

- Online-Plattform



Aufgabe 2 – Prozessmodelle

- Mars-Roboter



Aufgabe 2 – Prozessmodelle

Zusatzfolie

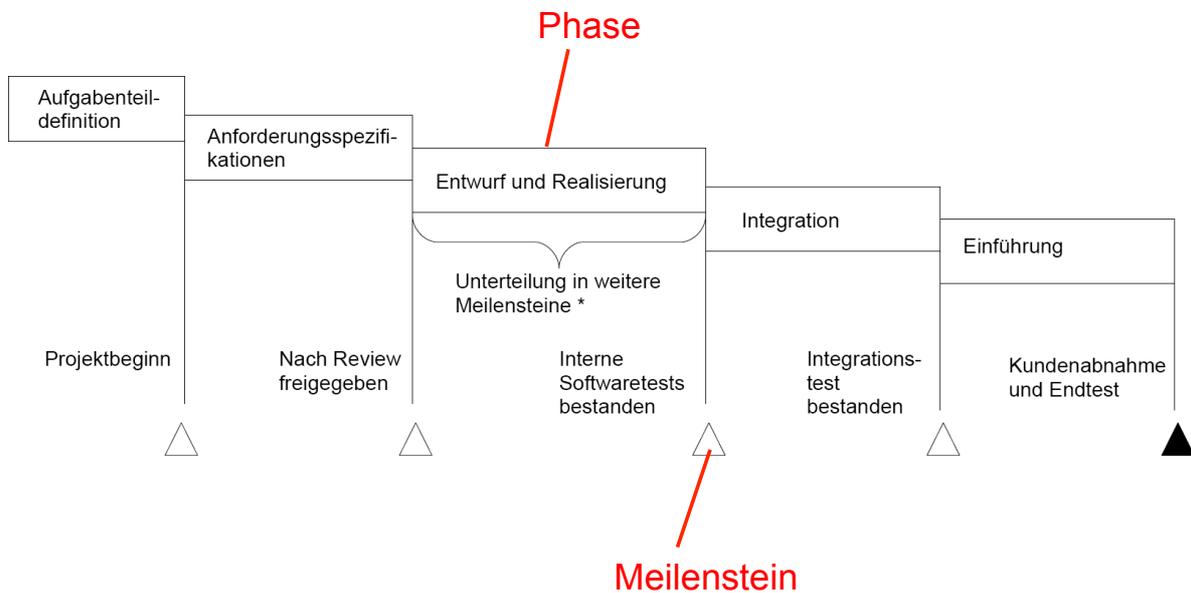
- Mars-Roboter
 - Gute, Vollständige Dokumentation wichtig!
(siehe auch Ariane 5 Fallstudie)
 - Zusammenarbeit mit Kunde **nicht nur** bei Agilen SE möglich
 - Welche Teillieferungen? Roboter funktioniert nicht mit der Hälfte

Agile SE als Prozesswahl eher problematisch...

Aufgabe 3 – Meilensteine

- Zweck
 - Den Prozess strukturieren (z.B. Lieferungen)
 - Den Fortschritt evaluieren
- Kriterien
 - Konsistent mit Antwort zu Aufgabe 2 b)?
 - Überprüfbare Meilensteine (z.B. durch Review, Tests)
 - Zeitliche Abfolge deutlich?

Aufgabe 3 – Meilensteine



Aufgabe 3 – Meilensteine

- Unterschied Meilenstein – Überprüfung
Anf. Spezifikation vorhanden – Durch Review überprüft

- Meilenstein Beispiel:

- Teillieferung 1 (Sitzplatzreservation) ✗
- Teillieferung 1 (Sitzplatzreservation) fertiggestellt (✗)
- Sitzplatzreservation funktioniert fehlerfrei (✗)
- Die Sitzplatzreservation Use Cases sind fehlerfrei ausführbar ✓
- Teillieferung 1 hat internen Abnahmetest bestanden ✓

Überprüfung: Die Komponente wird anhand der spezifizierten Use Cases getestet. Wenn alle Testfälle erfolgreich sind, hat die Komponente den Abnahmetest bestanden und der Meilenstein wurde erreicht

Aufgabe 4 – Prototypen

- Zweck
Frühes Erkennen und Lösen von Problemen
- **Explorativ**
Demonstration: Machbarkeit, Nützlichkeit (Projektakquisition)
Prototyp im engeren Sinn: Angemessenheit von Anforderungen, Tauglichkeit vorgesehener Lösungen (z.B. auch für GUIs)
- **Experimentell**
Labormuster, Untersuchung der Realisierbarkeit kritischer Systemteile
Bewertung von Entwurfsalternativen
- **Evolutionär**
Pilotsystem, entspricht einem Wachstumsmodell

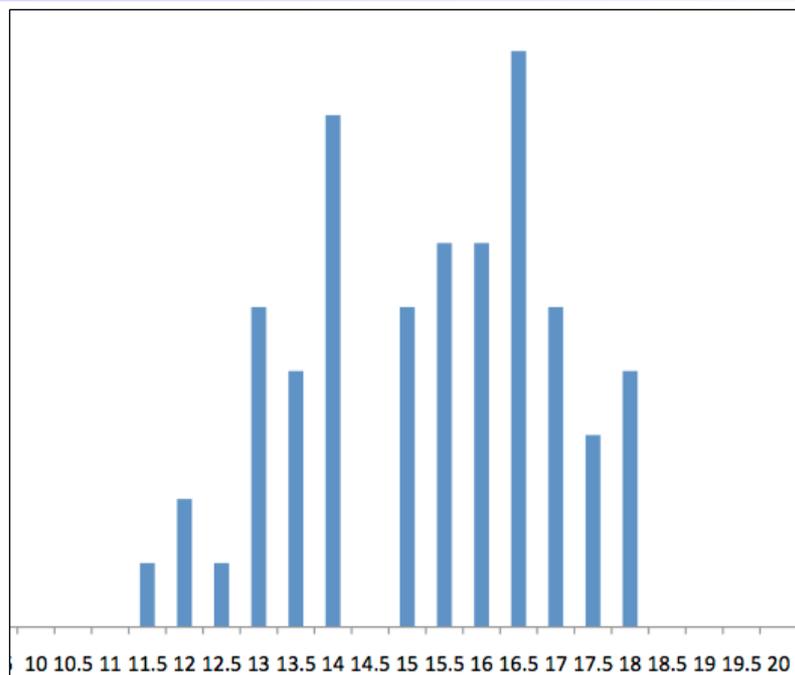
Aufgabe 4 – Prototypen

- Zweck
Frühes Erkennen und Lösen von Problemen
- **Explorativ**
Demonstration: Machbarkeit, Nützlichkeit (Projektakquisition)
Prototyp im engeren Sinn: Angemessenheit von Anforderungen, Tauglichkeit vorgesehener Lösungen (z.B. auch für GUIs)
Wegwerf-Prototypen
 - Können unsauber programmiert werden
 - Sinnvoll, wenn
Einsparung in Entwicklungskosten > Kosten Prototyp, wesentliche Reduktion eines gefährlichen Risikos
 - Brauchen Zeit und Geld
- **Experimentell**
Labormuster, Untersuchung der Realisierbarkeit kritischer Systemteile
Bewertung von Entwurfsalternativen
- **Evolutionär**
Pilotsystem, entspricht einem Wachstumsmodell
Kern des Systems
 - Saubere Programmierung, Dokumentation

Aufgabe 4 – Prototypen

- Prototypen im ergebnisorientierten Phasenmodell
 - Normalerweise keine evolutionären Prototypen

Übung 3 Resultate



Allgemeines

- Fragen zur Übung 3 oder Übung 4?
- Übung 4, Architekturstile: (~~UML~~) Diagramm
- Architekturstil, Einschränkung:
z.B. bei *Pipe and Filter*: Komponenten kennen ihre
Nachbarn nicht (Folie 52, Kapitel 5)