

# Software-Entwicklungs-Praktikum (SWEP) Einführung und Organisatorisches

Prof. Dr. Harald Störrle

Sommersemester 2009

## Formalia

SS 2009  
Software Entwicklungspraktikum  
0 – Organisatorisches – 2

- **Studierende mit Haupt- und Nebenfach Informatik**
  - 6 SWS, Diplom-Veranstaltung (keine ECTS-Punkte)

**Alle angemeldeten Studenten werden zur Veranstaltung zugelassen.**

- **Räume und Zeiten**

- **Dienstag 16.15 – 17.45 Rm 1.14**
- **Freitag 14.15 – 17.45 Rm 1.35**

- **Ablauf**

- **Gruppenarbeit, Plenum mit Ergebnispräsentation**
- **Plenum mit Vorträgen zu speziellen Themen (teilweise als Referat der Gruppen)**
- **In drei Stufen von je etwa 4 Wochen**
  - A – Anforderungen
  - B – Qualität
  - C – Präsentation

- **Accounts für das Projekt-Wiki, -Trac, -SVN etc. werden baldmöglichst zugeteilt.**

Name	Aufgaben	Kontakt
<b>Prof. Dr. Harald Störrle</b>	Leitung Gruppen 7, 8	Harald.Stoerrle@pst.ifi.lmu.de
<b>Prof. Dr. Rudolf Haggenmüller</b>	Unterstützung	-
<b>Dr. Shadi Al Dehni</b>	Architektur Gruppen 5, 6	aldehni@pst.ifi.lmu.de
<b>Gilbert Beyer</b>	Web/Wiki Gruppen 3, 4	Gilbert.Beyer@pst.ifi.lmu.de
<b>Christian Kroiß</b>	REFLECT-Expertise	-
<b>Laeith Raed</b>	Prozess/Werkzeuge Gruppen 1, 2	raed@pst.ifi.lmu.de

(c) 2009 Prof. Dr. H. Störrle, Uni München

\_\_\_\_\_ **LMU**  
**Ludwig** \_\_\_\_\_  
**Maximilians** \_\_\_\_\_  
**Universität** \_\_\_\_\_  
**München** \_\_\_\_\_

## Die Vision des REFLECT Media Environments

Prof. Dr. Harald Störrle

Sommersemester 2009

## Reflect System

Ein großes Display lockt Betrachter an, die von einer Videokamera aufgenommen werden. Die Bildanalyse berechnet die Position von Gesichtern und Händen der Betrachter.

Die Bildanalyse ergibt Signale für die Steuerung des Systems (z.B. Gesten).

## Reflect Media Player

Fehlt

Der Reflect Media Player (RMP) zeigt 3D-Szenen an (eventuell eingebettetem Video-Stream) und wird über das Netz ferngesteuert.

Kommandos vom Client lösen dreidim. geometrische Transformationen aus.

Mit dem Interaction Designer werden Interaktionen für REFLECT- und andere Applikationen entworfen.

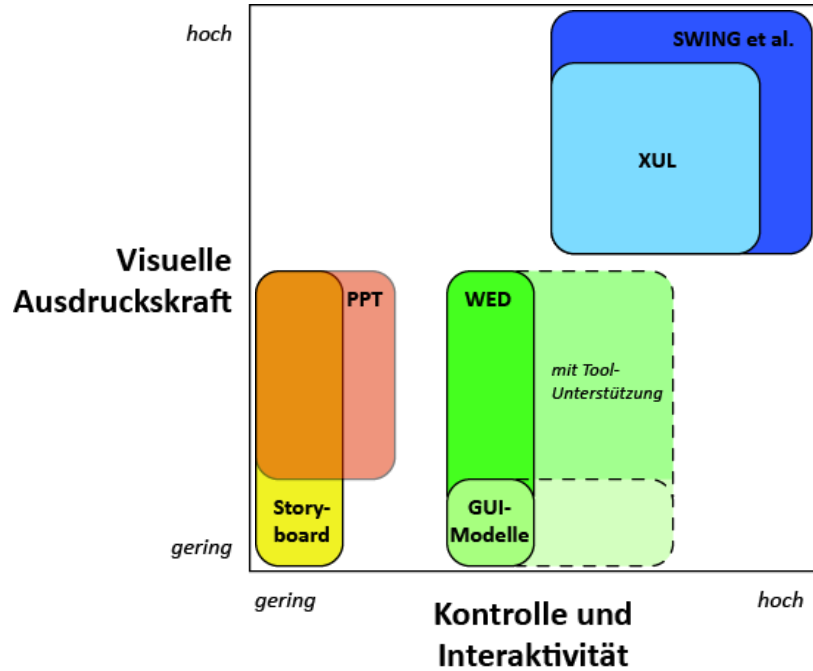
## Reflect Interaction Designer

Fehlt

# Reflect Interaction Designer: Die Vision

- **Zwischen den Interaction Designern und den Entwicklern bestehen große Unterschiede.**
  - Das betrifft die Herangehensweise und Werkzeuge, die jeweiligen Stärken und Schwächen, die Kultur, die Ziele und Schwerpunkte usw.
- **Tools unterstützen heute entweder das eine oder das andere.**
  - Storyboarding/Sketching-Werkzeuge eignet sich für HTML-Seiten und für den graphischen Grobentwurf.
  - GUI-Designer eignen sich für Fein-Entwurf und Code-Erzeugung einzelner Dialoge.
- **Der REFLECT Interaction Designer soll beide Bereiche verbinden, und einen harmonischen Übergang von Grob- über Feinentwurf zum Code erlauben.**
  - Dazu können Window-Event-Diagrams eingesetzt werden.

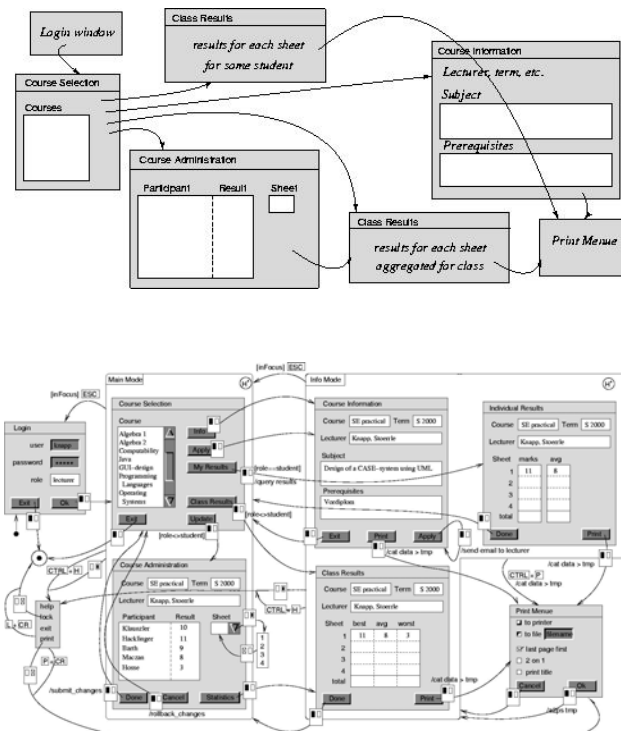
- GUI-Prototypen können grundsätzlich zwei Aspekte darstellen:
  - den Kontrollfluss / die Interaktivität, und
  - die visuelle Gestaltung.



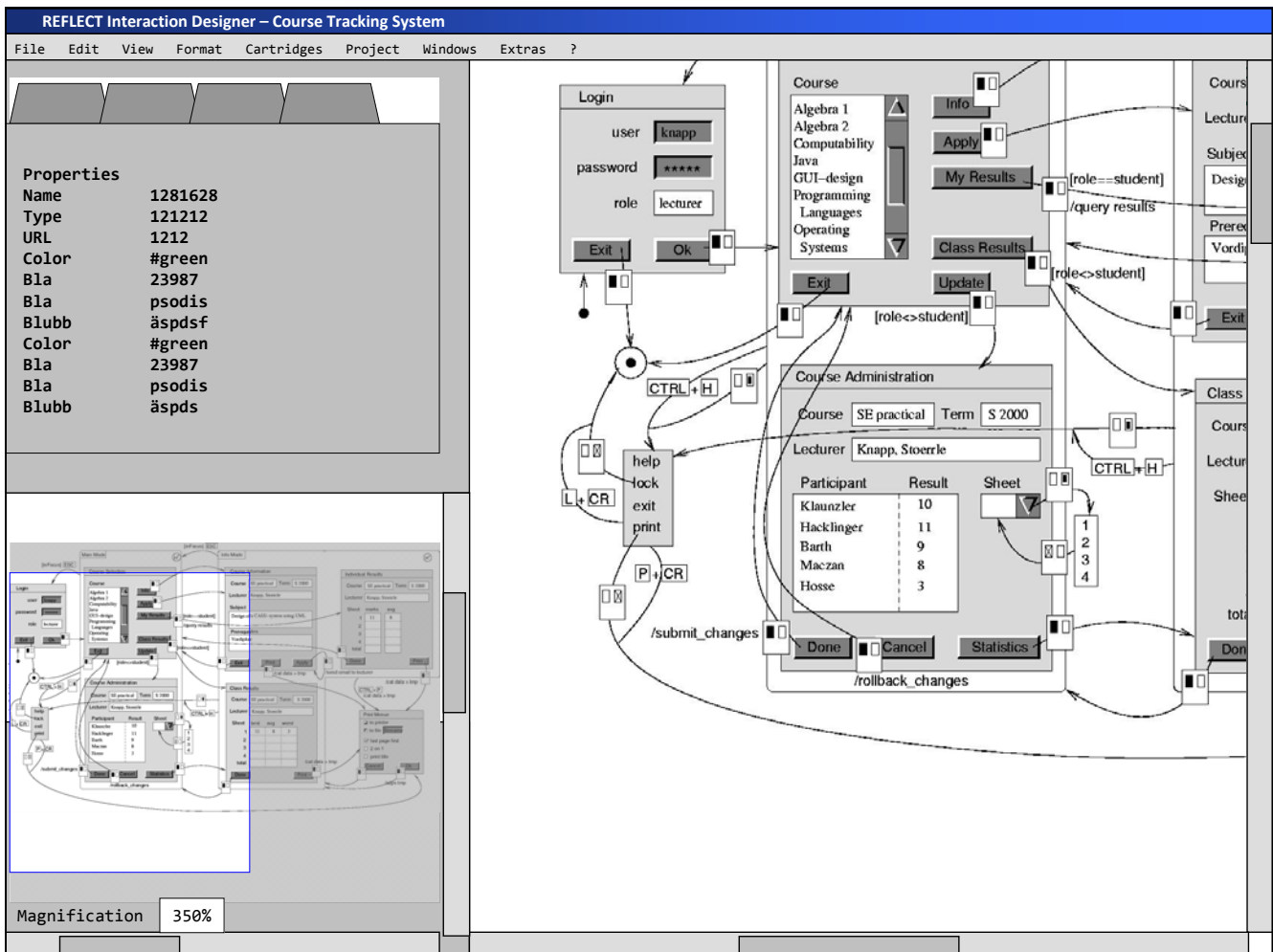
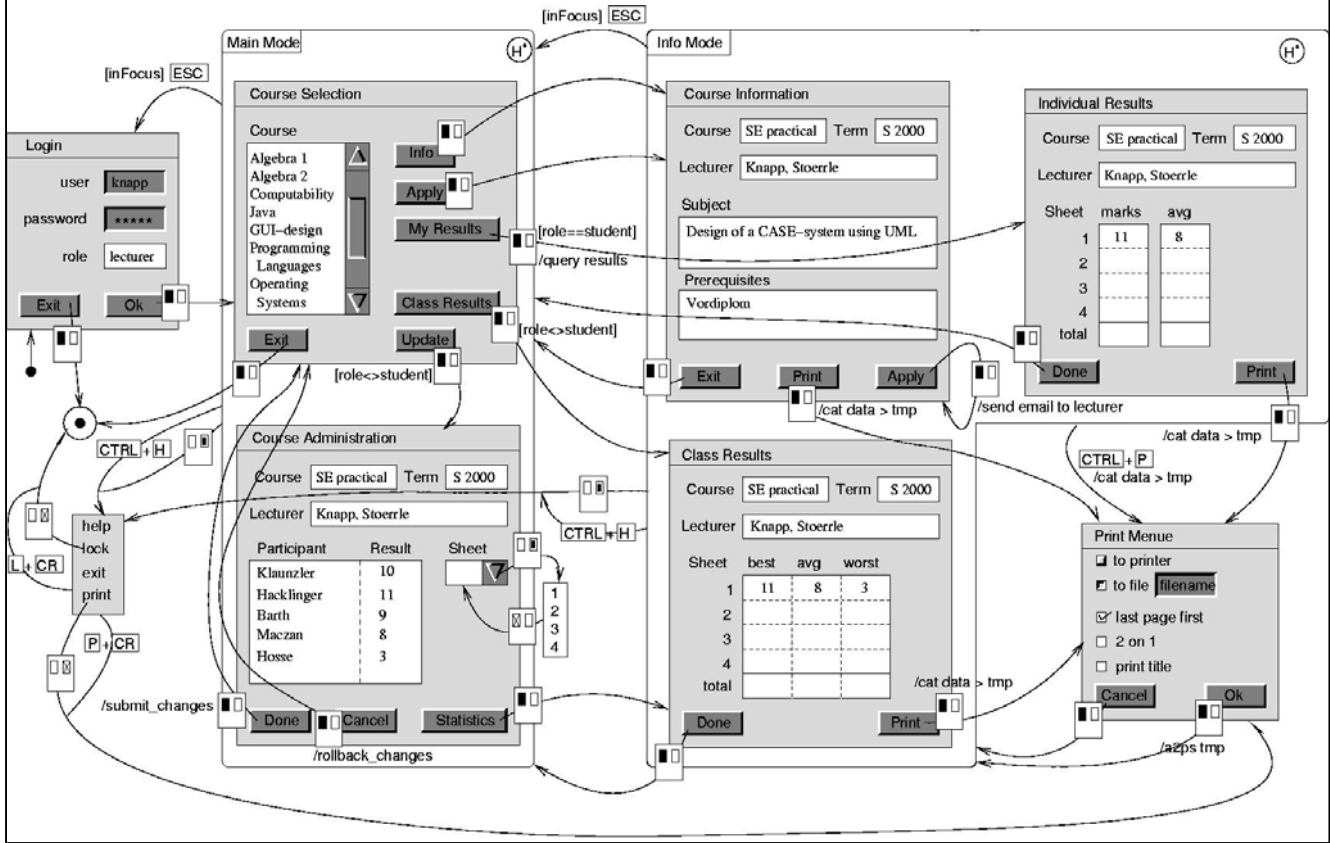
Mit visueller Ausdruckskraft sind die Möglichkeiten gemeint, graphische Elemente im Prototypen mit vertretbarem Aufwand darzustellen.

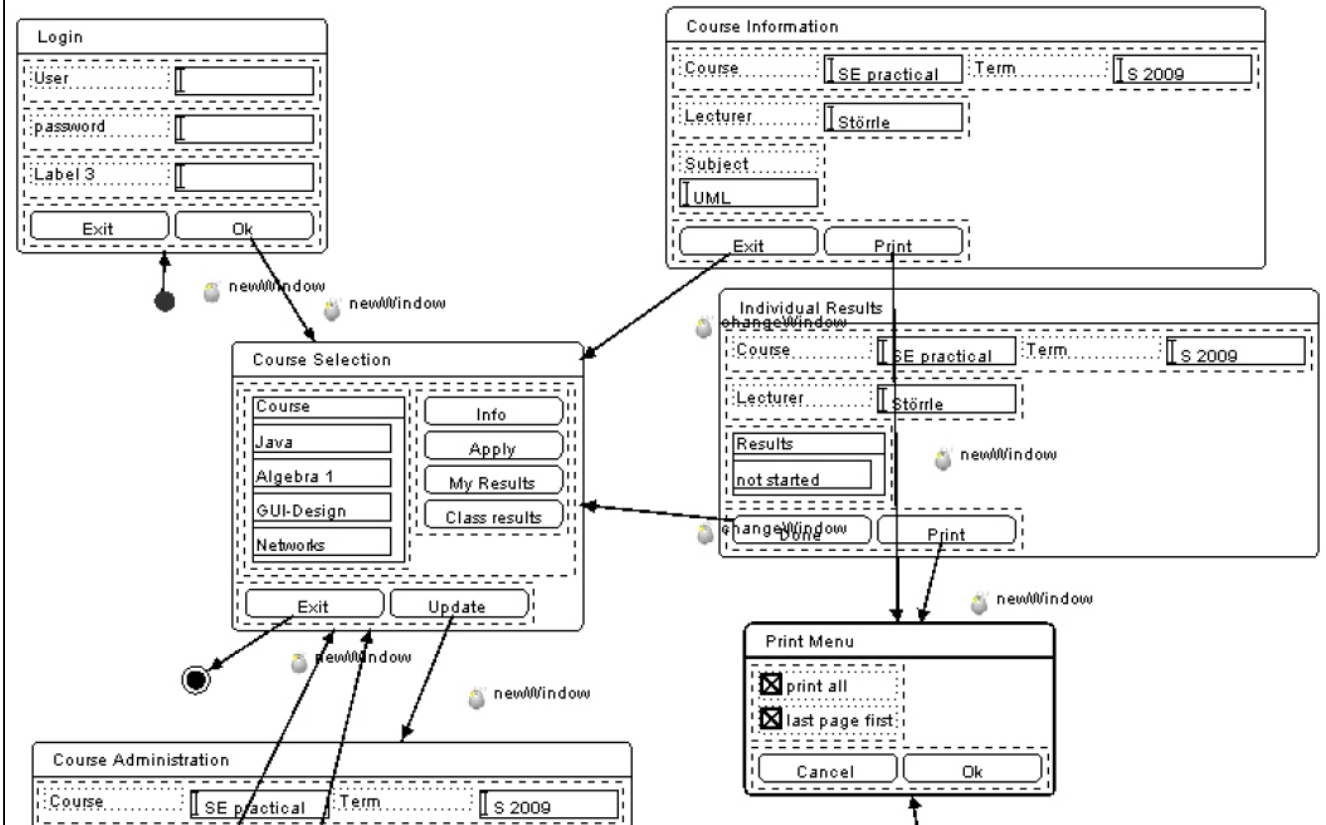
## Window-Event-Diagrams (WEDs)

- WEDs sind graphisch annotierte UML State Machines:
  - Zustände korrespondieren zu graphischen Elementen (Widgets, Fenster, ...);
  - Übergänge korrespondieren zu Benutzeraktionen (Klicks, Tastatureingaben, ...).
- Damit ist ein WED gleichzeitig eine Spezifikation für die Entwickler und ein Entwurf für den Interaction Designer.
- WEDs können wahlweise eher grob oder eher fein sein, sie können eher softwaretechnisch oder graphisch oder prozessorientiert sein.



# Ein komplexes WED



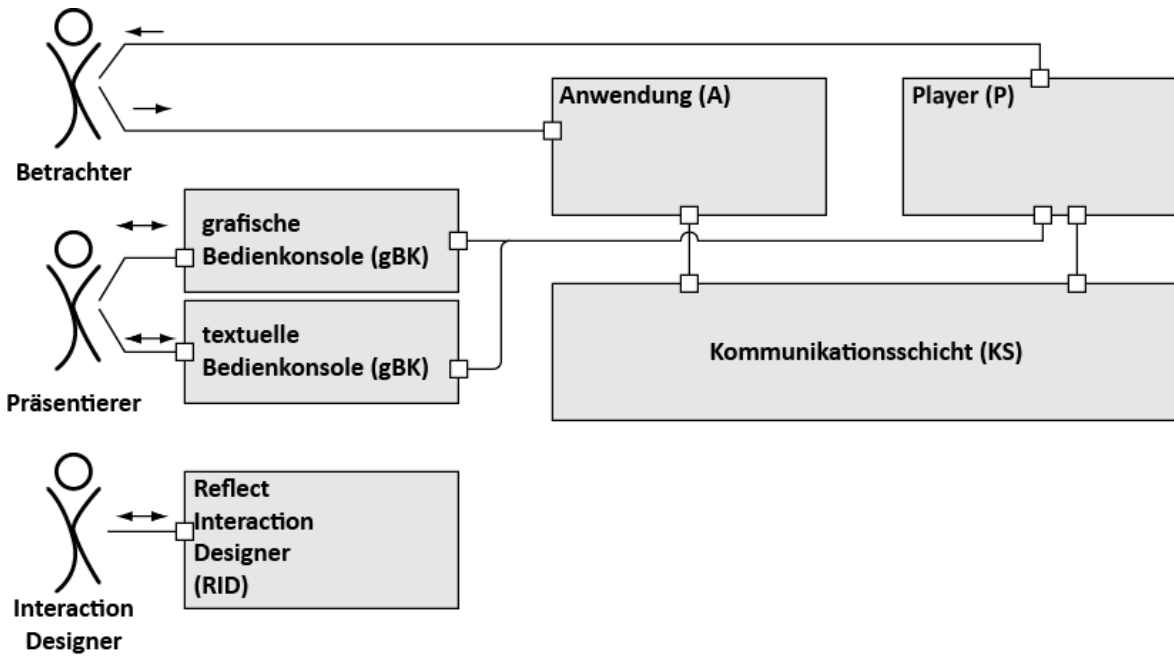


## Die Software-Architektur des REFLECT Media Environments

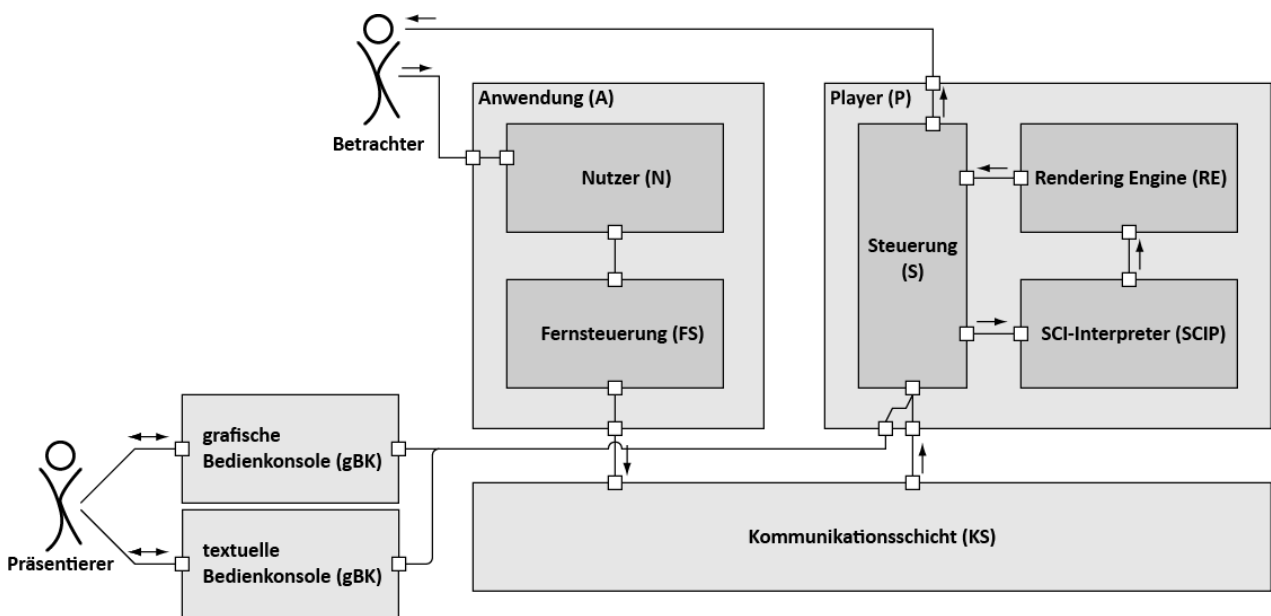
Prof. Dr. Harald Störle

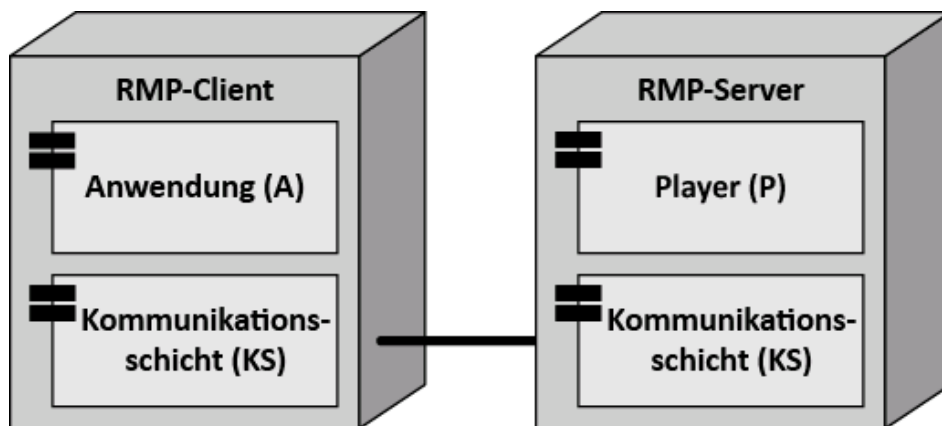
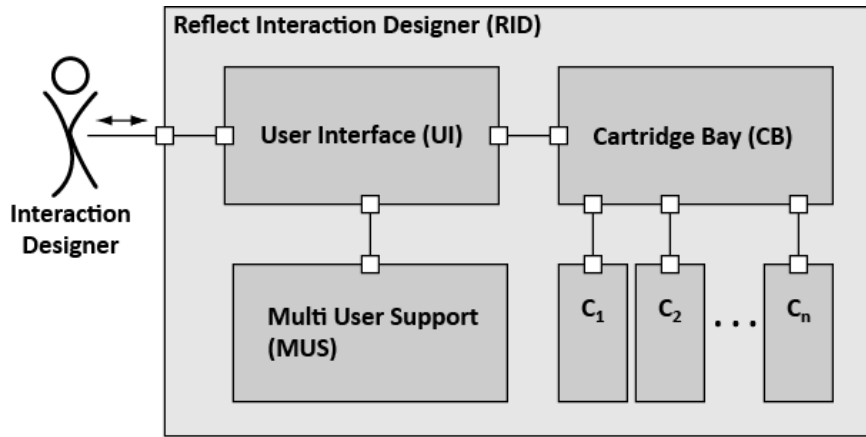
Sommersemester 2009

# Softwarearchitektur RME



# Softwarearchitektur RMP







Laeith Raed

- 1. Anforderungen**
  - Prozess für Anforderungen, Änderungen und Fehler
  - Sichtung und Verwaltung der Anforderungen

- 2. Test**
  - Code- und Architektur-Reviews
  - Last-/Stress-/Stabilitätstests
  - Demo
  - Abnahmetest

- 3. Dokumentation**
  - Ablagestruktur für Dokumente
  - Vorgaben zu Gliederung und Kapitelinhalten, Richtlinien für Texte und Abbildungen
  - Redaktion

- 4. Test**
  - Vorgaben für Komponententest
  - Kontrolle der Einhaltung durch Review
  - Planung und Durchführung des (fortlaufenden) Integrationstests

Shadi Al Dehni

- 5. Konfigurationsmanagement**
  - Versions- und Konfigurationsverwaltung
  - Build-Prozess
  - Systemintegration

- 6. Architektur**
  - Auswahl und Einhaltung von Codierungsrichtlinien
  - JavaDoc, Refactoring, Patterns
  - Einsatz von Bibliotheken

- 7. User Interface**
  - Benutzbarkeitsrichtlinien
  - Analytische und konstruktive Methoden im Interaction Design
  - Usability Testing

- 8. Datenstrukturen**
  - VRML, Szenegraph
  - WED-Datenstruktur
  - Datenmodellierung

Harald Störrle

Gilbert Beyer

# Gruppeneinteilung

30?

Komponente(n)	Gruppe	Betreuer	Mitglieder		
Fernsteuerung, Bedienkonsolen	1	Raed	1	9	17
Kommunikationsschicht (Sender & Empfänger)	2	Raed	2	10	18
Steuerung	3	Beyer	3	11	19
SCI-Interpreter	4	Beyer	4	12	26
Rendering Engine	5	Al Dehni	5	21	33
Cartridge Bay, Gesamtstruktur	6	Al Dehni (Störrle)	6	14	22 28
User Interface Multi User Support	7	Störrle	7	15	29 32
Cartridges	8	Störrle	8	16	27 25

- **Anforderungen werden zunächst tabellarisch/textuell beschrieben. Das Schema besteht aus folgenden Rubriken.**
  - ID
  - Name
  - Beschreibung
  - Subsystem
  - Art
  - Priorität
  - Teil von
  - Erfüllungskriterium
  - Anmerkung
  - Aufwand
  - Status
  - Kommentar

- 1. SWEP-Anfo-Prozess und Werkzeugunterstützung vorbereiten**
- 2. RMP-Anforderungen konsolidieren**
- 3. SWEP-Ablagestruktur und Glossar vorbereiten**
- 4. RMP-Anforderungen nach UCs umsetzen**
- 5. Vortrag „KM-Einführung“ und KM-Struktur für SWEP vorbereiten**
- 6. Reengineering und Evaluierung Prototyp**
- 7. RID-Anforderungen konsolidieren und mit Prototyp abgleichen**
- 8. RID-Anforderungen nach UCs umsetzen (Unterstützung Grp. 6, 7)**

1. Vortrag „Qualitätssicherung von Anforderungen“ vorbereiten
2. Vortrag „Review und Test“ vorbereiten
3. Dokumente „Anforderungen und Anwendungsfälle“ erstellen
4. Vortrag „Unit-Test“ und Unit-/Integrations-Tests vorbereiten
5. KM-Einführung / KM für SWEP vorbereiten
6. Evaluierung von Bibliotheken, Architekturvorschlag
7. Vortrag „GUI-Richtlinien“ vorbereiten, GUI-Entwurf RID
8. Datenstruktur WED, Datenstruktur SG



**Prof. Dr. Harald Störrle**

Programmierung und softwaretechnik  
Institut für Informatik  
Uni München  
Oettingenstr. 52  
D-80538 München

mgm technology partners GmbH  
Frankfurter Ring 105a  
80807 München

[Harald.Stoerrle@pst.ifi.lmu.de](mailto:Harald.Stoerrle@pst.ifi.lmu.de)  
[www.pst.ifi.lmu.de/~stoerrle](http://www.pst.ifi.lmu.de/~stoerrle)  
[www.mgm-tp.com](http://www.mgm-tp.com)