

Einführung in die Softwaretechnologie

Wintersemester 2013

Dr. Günter Kniesel
Institut für Informatik III
Römerstr. 164, D-53117 Bonn
gk@cs.uni-bonn.de

<http://sewiki.iai.uni-bonn.de/teaching/lectures/se/2013/>

Kapitel 0. „Organisatorisches“

Stand: 14.10.2013

Studiengänge / Prüfungen

Tutorien

Infrastruktur

Informatik-DPO 1998

- B-Prüfung
 - ◆ in Kombination mit anderer 4-stündigen Veranstaltung aus dem B-Bereich
- Oft gewählt
 - ◆ Fächerkombination
 - ⇒ SWT + IS
 - ◆ Mögliche Prüfer:
 - ⇒ Prof. Dr. R. Manthey + Dr. Kniessel
 - ⇒ Prof. Dr. R. Manthey + ~~Prof. Dr. A. B. Cremers~~
 - ⇒ ~~Prof. Dr. A. B. Cremers (beides)~~

Bachelor und Informatik-DPO 2003

- Modulprüfung
 - ◆ Schriftliche Klausur
 - ◆ Schriftliche Nachklausur
 - ◆ Schriftliche Nach-Nachklausur

Voraussetzung zur Prüfungsteilnahme

- Für Bachelor Studierende
 - ◆ Vorlesung 015 – Objektorientierte Programmierung – bestanden!
- Nebenfächler
 - ◆ Was immer Ihre Prüfungsordnung bestimmt...

- Basisliteratur
 - ◆ Brügge, Dutoit:
“Object oriented Software Engineering – Conquering Complex and Changing Systems”, Prentice Hall, Zweite Auflage.
 - ◆ bzw. Deutsche Ausgabe des Buches.

- Weitere
 - ◆ <http://sewiki.iai.uni-bonn.de/teaching/lectures/se/2013/>

- Was
 - ◆ 1 Stunde
 - ◆ 1 Betreuer ⇔ 6-7 Studenten
 - ◆ Abgabe via SVN
- Warum
 - ◆ Praktische Anwendung des Vorlesungsstoffes
 - ◆ Feedback zu **Ihren** Lösungen
- Wann
 - ◆ Ab nächster Woche
 - ◆ Ausgabe der Übungszettel: Montag vor 10:00 Uhr
 - ◆ Abgabe der Lösungen: Bis Sonntag 23:59 (im SVN Repository)

Tutorien-Zeiten

Zeiten	Montag	Dienstag	Mittwoch	Donnerstag
7:00	☒ 0 👤 2 +	☒ 0 👤 2 +	☒ 0 👤 2 +	☒ 0 👤 1 +
8:00	☒ 0 👤 4 +	☒ 0 👤 2 +	☒ 0 👤 0 +	☒ 0 👤 1 + <div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p>+ Gruppe 17</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 18</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 19</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> </div>
9:00	☒ 0 👤 4 +	☒ 0 👤 2 +	☒ 0 👤 0 +	☒ 0 👤 1 + <div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p>+ Gruppe 20</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 21</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 22</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> </div>
10:00	☒ 0 👤 0 +	☒ 0 👤 4 +	☒ 0 👤 3 +	☒ 0 👤 4 + <div style="width: 30%;"> <p>+ Gruppe 3</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 23</p> <p>☒ keiner</p> <p>👤 0/3</p> </div>
11:00	☒ 0 👤 0 +	☒ 0 👤 4 +	☒ 0 👤 3 +	☒ 0 👤 4 + <div style="width: 30%;"> <p>+ Gruppe 4</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 24</p> <p>☒ keiner</p> <p>👤 0/3</p> </div>
12:00	☒ 0 👤 4 +	☒ 0 👤 1 +	☒ 0 👤 5 +	☒ 0 👤 1 +
13:00	☒ 0 👤 4 +	☒ 0 👤 1 +	☒ 0 👤 5 +	☒ 0 👤 1 +
14:00	☒ 0 👤 3 + <div style="width: 30%;"> <p>+ Gruppe 5</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 6</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 7</p> <p>☒ keiner</p> <p>👤 0/3</p> </div>	☒ 0 👤 3 + <div style="width: 30%;"> <p>+ Gruppe 8</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 9</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 10</p> <p>☒ keiner</p> <p>👤 0/3</p> </div>	☒ 0 👤 2 + <div style="width: 30%;"> <p>+ Gruppe 11</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 13</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 14</p> <p>☒ keiner</p> <p>👤 0/3</p> </div>	☒ 0 👤 2 +
15:00	☒ 0 👤 3 + <div style="width: 30%;"> <p>+ Gruppe 1</p> <p>☒ keiner</p> <p>👤 0/3</p> </div>	☒ 0 👤 4 + <div style="width: 30%;"> <p>+ Gruppe 2</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 12</p> <p>☒ keiner</p> <p>👤 0/3</p> </div>	☒ 0 👤 2 + <div style="width: 30%;"> <p>+ Gruppe 15</p> <p>☒ keiner</p> <p>👤 0/3</p> </div> <div style="width: 30%;"> <p>+ Gruppe 16</p> <p>☒ keiner</p> <p>👤 0/3</p> </div>	☒ 0 👤 3 +
16:00	☒ 0 👤 3 +	☒ 0 👤 2 +	☒ 0 👤 1 +	☒ 0 👤 2 +

Online-Anmeldung zu Tutorien

1. URS-Anmeldung

- ◆ <https://sewiki.iai.uni-bonn.de/teaching/lectures/se/2013/anmeldung>

2. Nicht-Verfügbarkeit angeben

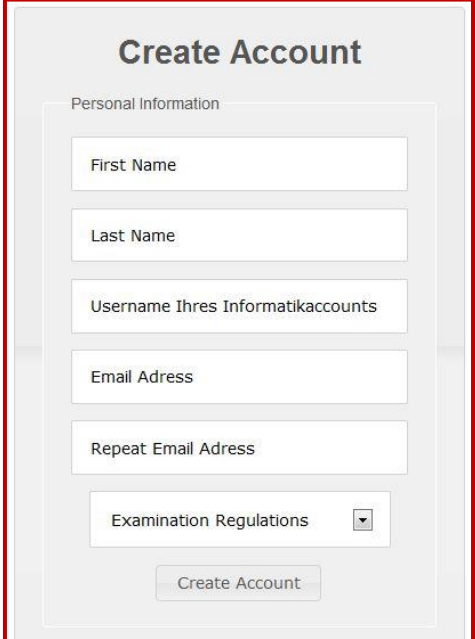
- ◆ Reihenfolge der Anmeldung ist **nicht** wichtig
- ◆ Sie können sich beliebig oft umentscheiden
- ◆ ... bis **Mittwoch, 16.10.2013, 23.00 Uhr**

3. Benachrichtigung

- ◆ Donnerstag nachmittag / abend
- ◆ E-Mail: Gruppe und SVN-Kennung

3. SVN-Anmeldung

- ◆ Innerhalb 2 Tagen



The image shows a 'Create Account' form with the following fields:

- Personal Information
- First Name
- Last Name
- Username Ihres Informatikaccounts
- Email Address
- Repeat Email Address
- Examination Regulations (dropdown menu)
- Create Account button

A red box surrounds the form, and a red arrow points from the URL in the first list item to the right side of the form.

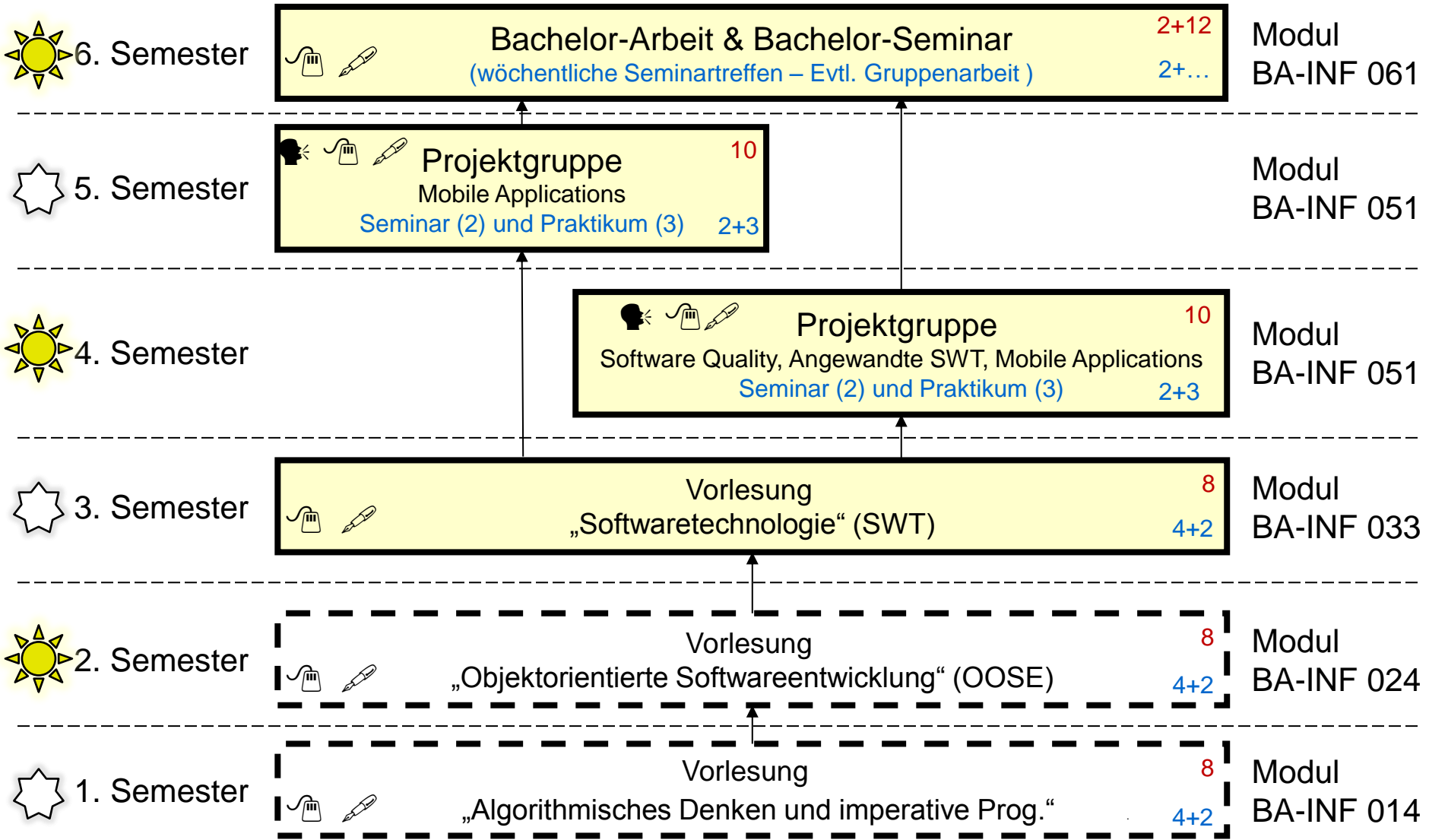
Übungsblatt 0

- <http://sewiki.iai.uni-bonn.de/teaching/lectures/se/2013/fohlen>
- Aufgabe: Voraussetzungen zur Übungsteilnahme schaffen
 - ◆ Anmeldung zur Tutoriumsgruppe
 - ◆ Aktivierung des Altbau Windows Accounts
 - ◆ Eclipse-Installation
 - ◆ Aktivierung des SVN-Accounts
 - ◆ SVN-Zugang testen (Gedicht)
- Abgabe im ersten Tutorium an Ihren Tutor

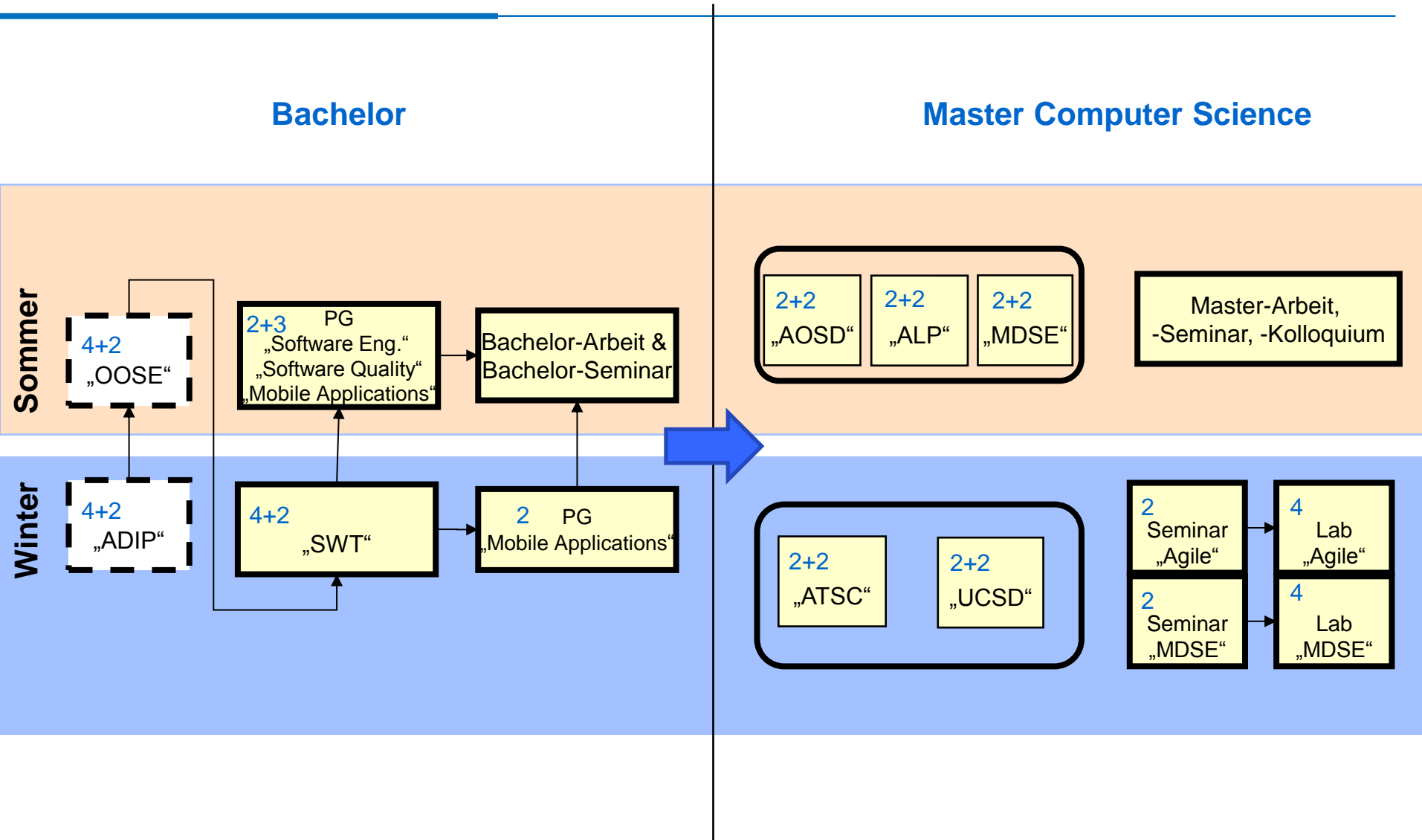
Ausblick

Software Engineering Curriculum
Unmittelbare Nachfolgeveranstaltungen

SE-Curriculum im Informatik-Bachelor-Studiengang



Mögliche Gesamt-Studienverläufe



Unmittelbare Anschluss-Veranstaltungen

- Projektgruppen
 - ◆ Praktische Softwaretechnologie → Sommersemester
 - ◆ Mobile Applikationen → Sommersemester + Wintersemester
- Teilnahme
 - ◆ Bachelorstudiengang
 - ◆ Diplomstudiengang
- Anmeldung
 - ◆ Infos demnächst im SE-Wiki <http://sewiki.iai.uni-bonn.de/teaching/>

Sommersemester: Projektgruppe „Angewandte Softwaretechnologie“

- In kleinen Gruppen können **eigene** Projektideen betreut bearbeitet werden
- Praktische Anwendung der in Vorlesung Softwaretechnologie behandelten Technologien, Werkzeuge und Entwicklungsprozesse

Vorbesprechung:

letzte Vorlesungswoche des Wintersemester

Voraussetzung:

Vorlesung Softwaretechnologie

Betreuer: nonnen@iai.uni-bonn.de



Sommer- und Wintersemester: PG „Mobile and Social Applications“

- Softwareentwicklung auf iPhone-OS und Android
- Mobile ortsbasierte Anwendung, verteilte Kollaboration, soziale Interaktion

Vorbesprechung:

letzte Vorlesungswoche des Wintersemester

Voraussetzung:

Vorlesung Softwaretechnologie

Betreuer: muegge@iai.uni-bonn.de, bihler@iai.uni-bonn.de, vonzeschau@cs.uni-bonn.de



Kapitel 1

Software Engineering – Überblick

– Stand: 14.10.2013 –

Kosten

- Softwarekosten übersteigen oft Systemkosten
 - ◆ Beispiel: PC
- Softwarewartung übersteigt Entwicklungskosten
 - ◆ Bis zu 80% der Gesamtkosten

Qualität

- Schlechte Software schadet
 - ◆ Direkte Schäden
 - ◆ Unzufriedene Benutzer
 - ◆ ...

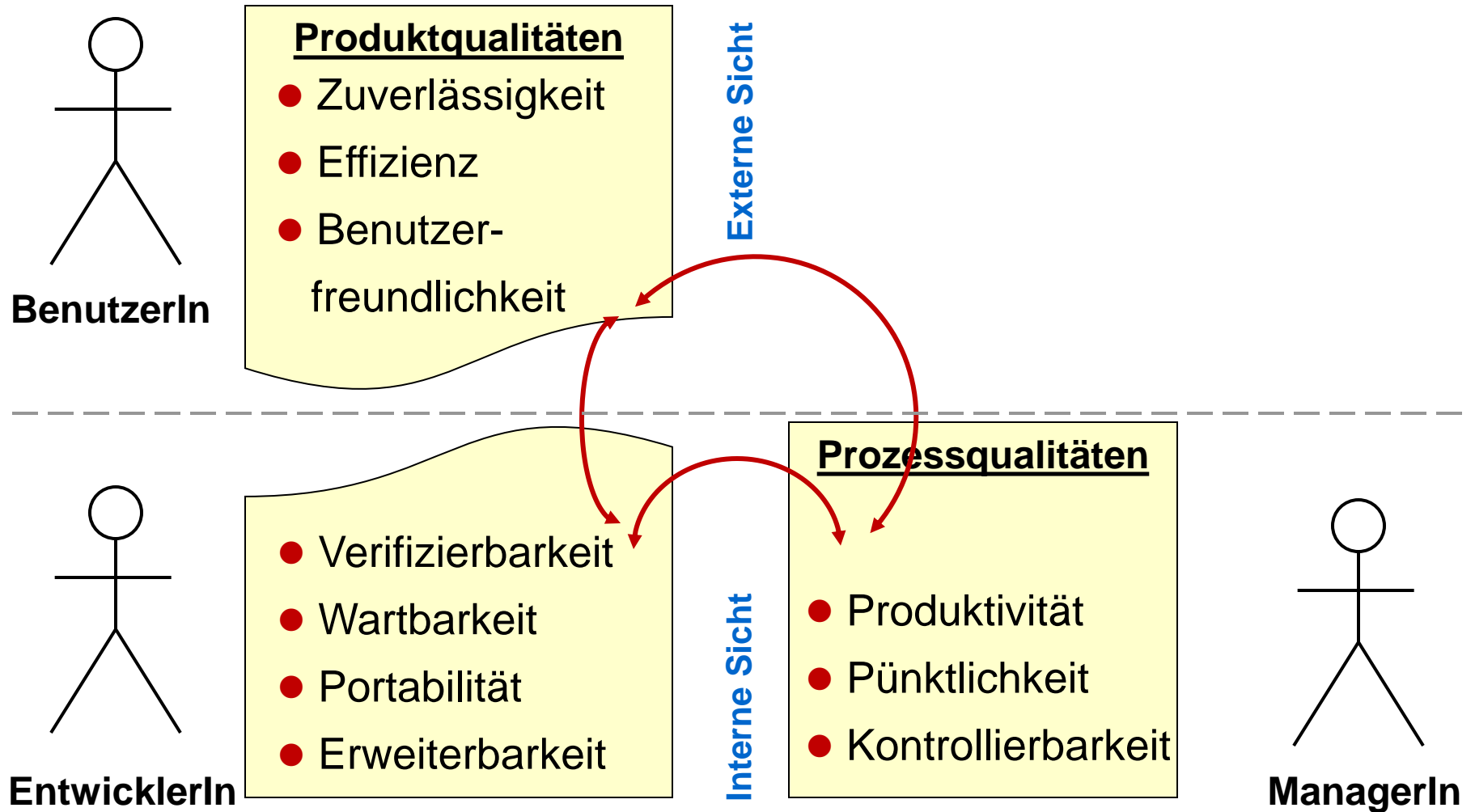
→ **Kosteneffektive Entwicklung** von **qualitativ hochwertiger Software**

Fehlleistungen (Beispiele)

- Zu „enger“ Weltausschnitt
 - ◆ Im Jahre 1992 erhielt Mary (geb. 1888) eine Einladung in den Kindergarten
- Fehlverhalten
 - ◆ Gepäcksystem am Flughafen von Denver
 - ◆ beschädigte Koffer
 - ◆ 3,2 Mio. \$ über dem Budget
 - ◆ 16 Monate verspäteter Start mit zum größten Teil manuellem System
- Unnötig komplexe Lösung
 - ◆ C-17 Transportflugzeug ▶ 19 Computer, 6 Programmiersprachen
 - ◆ 500 Mio \$ über dem Budget
- Interface-Missbrauch
 - ◆ Wissend, dass es ein System gab, das die Abfahrt des Zuges mit offenen Türen verhinderte, fixierte ein Zugführer den Startknopf mit Klebeband in Startstellung.
 - ◆ Als er den Zug verließ, um eine klemmende Tür zu schließen, fuhr der Zug ohne ihn los (als die Tür nicht mehr klemmte).

Er fand dies eine besonders clevere Art sicherzustellen, dass der Zug sofort losfuhr sobald die letzte Tür geschlossen war.

Software-Qualitäten: Verschiedene Sichten



Herausforderungen

Komplexität

- Viele Funktionen
- Viele Ziele ▶ Interessenskonflikte
- Viele Komponenten ▶ Komplexität der Zusammenstellung
- Viele Teilnehmer ▶ Komplexität der Kommunikation

Keine Einzelperson kann ein ganzes System verstehen

Änderung

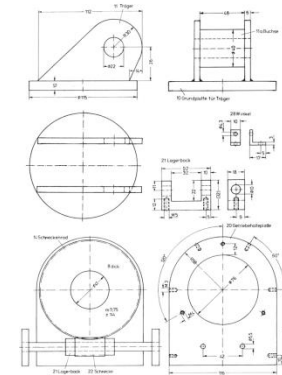
- **Welt** ▶ Gesetzgebung, Geschäftsabläufe, ...)
- **Der modellierte Ausschnitt** ▶ mehr Funktionalität, Systeme integrieren, ...
- **Die Implementierungstechnologie** ▶ neue Sprachen, Komponenten, ...
- **Das Team** ▶ Personen gehen, neue kommen hinzu, ...
- **Das Management** ▶ neue Geschäftsausrichtung, ...

Software ist ständiger Änderung unterworfen!

Andere Ingenieurwissenschaften

Domäne

- Klar definierte Probleme
- Ein Produkt muss *gebaut* werden
- Hohe Qualitätsanforderungen



Methoden

- Systematische Verfahren und ihre disziplinierte Anwendung
- Standardschnittstellen, -komponenten und -prozesse
- Wissen um verfügbare Komponenten
- Empirische Methoden zur Bewertung von Lösungen



Software Engineering ist anders

• Andere Ingenieursbereiche

- ◆ Herstellung bestimmt die Endkosten
- ◆ Änderungen sind weniger häufig
 - ⇒ 2000 Jahre von der Euklidischen zur nicht-Euklidischen Geometrie
- ◆ Änderungen sind möglich, aber sehr teuer
 - ⇒ Redesign wird gründlich durchdacht
 - ⇒ Auswirkungen werden genau analysiert

• Software Engineering

- ◆ Herstellung ist eine einfache Duplizierung
- ◆ Änderungen geschehen dauernd
 - ⇒ Manchmal innerhalb von Stunden (Kunde hat seine Meinung geändert)
- ◆ Änderungen sind einfach
 - ⇒ Kein durchdachtes Redesign
 - ⇒ Auswirkungen werden nicht ausreichend bedacht



Andauernde Änderungswünsche zusammen mit der Leichtigkeit, Änderungen durchzuführen, führen zu ungenügend überdachten, ad-hoc „Lösungen“.



Professionelle Softwareentwicklung

- **Werkzeuge**
 - ◆ Konzepte, Sprachen, Systeme, ...
- **Methoden**
 - ◆ Wie setzen wir die Werkzeuge ein?
- **Prozesse**
 - ◆ Wie organisieren wir das ganze?

Übersicht: Themenbereiche

Teamarbeit und Änderungen

- ◆ Configuration Management
- ◆ Issue Management
- ◆ Rationale Management
- ◆ Project Management

Objektorientierte Modellierung

- ◆ OOP → UML
- ◆ Design Patterns
- ◆ Refactoring

SE Aktivitäten („Workflows“)

- ◆ Anforderungserhebung
- ◆ Analyse
- ◆ Systemdesign
- ◆ Objektdesign
- ◆ Testen

Softwareprozess

- ◆ Wasserfall-Modell
- ◆ V-Modell
- ◆ Spiral-Modell
- ◆ Unified Process
- ◆ Agile SW-Entwicklung