

# Software-Projekt

Prof. Dr. Rainer Koschke

Fachbereich Mathematik und Informatik  
Arbeitsgruppe Softwaretechnik  
Universität Bremen

Wintersemester 2008/09

# Überblick I

- 1 Vorbemerkungen

- 1 Vorbemerkungen
  - Ziele und Inhalt
  - Angestrebte Resultate
  - Aktivitäten der Software-Entwicklung
  - Ablauf
  - Zeitplan
  - Anmeldung
  - Scheinbedingungen
  - Projektplan
  - Vorstellung der Aufgabe
  - Kontaktdaten
  - Ressourcen
  - Lehrbücher

Einige dieser Folien basieren inhaltlich auf Skripten von . . .

- Prof. Dr. Jochen Ludewig (2003), Universität Stuttgart
- Prof. Dr. Susanne Maaß (2001), Universität Bremen
- Prof. Dr. Karl-Heinz Rödiger (2004), Universität Bremen
- Dr. Andreas Winter (2003), Universität Koblenz

Ich danke für deren freundliche Genehmigung.

## Primäre Ziele:

- Rüstzeug für die erfolgreiche Durchführung Ihres Software-Projekts vermitteln
  - Modell für zukünftige ähnliche Projekte bieten
- Praktikum und Vorlesung bilden eine Einheit

## Primäre Ziele sind **nicht**:

- Vollständige Darstellung aller Themengebiete der Softwaretechnik<sup>1</sup>
- Vermittlung von spezifischen Kenntnissen für die Entwicklung des Anwendungssystems X

---

<sup>1</sup>Wird im Hauptstudium nachgereicht.

## Szenario:

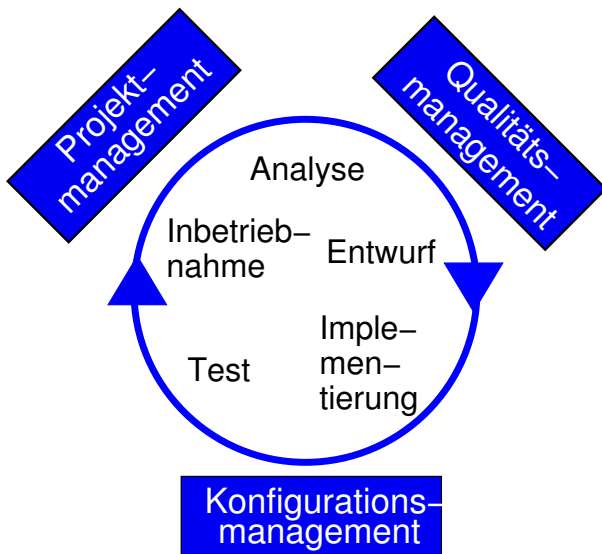
- Erstellung eines großen Softwaresystems
  - hier: mehrere Mitarbeiter über einen langen Zeitraum
  - hier **nicht**: ein „Freizeit“-Programmierer
- für einen Auftraggeber
  - hier: Individualsoftware
  - hier **nicht**: Standardsoftware
- im Rahmen eines Projekts
  - hier: einmalige Zielverfolgung
  - hier (eher) **nicht**: Weiterentwicklung existierender Software  
(aber: die Software wird von uns später weiterentwickelt; Wartbarkeit ist erklärtes Ziel)

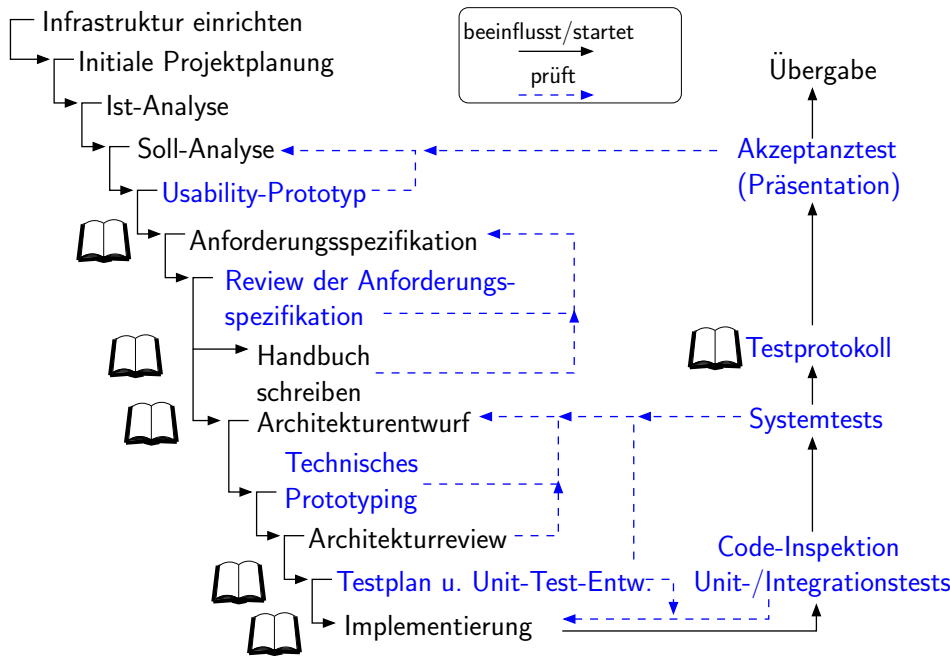
# Angestrebte Resultate

- Softwaretechnik ist nicht nur Programmieren.
- Softwaretechnik ist auch Programmieren.
- Software-Entwicklung produziert Dokumente.
- Der Quellcode ist **ein** Dokument **unter vielen**.

Sie sollen nicht nur ein richtiges System bauen.  
Sie sollen ein System auch richtig bauen.

# Aktivitäten bei der Softwareentwicklung



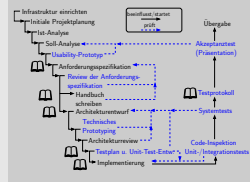



2008-10-26

# Software-Projekt

Vorbemerkungen

Ablauf



Aktivitäten in schwarzer Schrift sind konstruktiv; Aktivitäten in blauer Schrift dienen der analytischen Qualitätssicherung. Das Symbol  kennzeichnet abzugebende Dokumente, die Produkt einer Aktivität sind.

<b>24.11.2008</b>	<b>Initialer Projektplan</b>
<b>12.1.2009</b>	<b>Anforderungsspezifikation mit Angebot</b>
12.-30.1.2009	Review der Anforderungsspezifikation
vorauss. 9-13.2.2009	Vorlesung <i>Datenbankgrundlagen</i>
vorauss. 9-20.3.2009	Prüfung zur Vorlesung
<b>16.3.2009</b>	<b>Architkturentwurf</b>
6.-13.4.2009	Walkthrough des Architkturentwurfs
<b>20.4.2009</b>	<b>Testplan mit Unit-Tests</b>
Mai/Juni 2009	Code-Inspektion
<b>10.7.2009</b>	<b>Endabgabe<sup>2</sup></b>
15-17.7.2009	Abschlusspräsentationen

Vorläufige Teilergebnisse müssen Sie evtl. in den Tutorien vorher abgeben.

---

<sup>2</sup>Implementierung (Source+lauffähig), Dokumentation, Whiteboxtests, Testprotokoll, Dokumentation der Abweichungen von Spezifikation und Architektur

Tutorienwahl ab sofort bis zum bis 30.10., 23.59 Uhr<sup>3</sup>  
(erstes Tutorium am 3.11.)

Web-Seite zur Anmeldung:

<http://www.informatik.uni-bremen.de/st/mems/courses.php?lng=en>

- gruppenweise anmelden
- Gruppengröße 6 Personen
- Gruppennamen sorgfältig auswählen
- ausländische Studierende integrieren

---

<sup>3</sup>Ortszeit Bremen

<http://www.informatik.uni-bremen.de/st/Lehre/swp/scheinbedingungen.html>

# Abgabe des Projektplans

- abzugebender initialer Projektplan umfasst Aufgaben, Teilnehmer, Rollen, Verantwortlichkeiten, Pflichten, Risiken und deren Behandlung
- danach: inkrementelle Fortführung
  - kann abgegeben werden für Feedback vom Tutor, muss aber nicht
  - raten wir aber dringend an
  - Verantwortlicher ist z.B. Phasenleiter für die zu planende Phase
  - erste Woche jeder Phase dient Einarbeitung und Planung
  - Granularität der Arbeitspakete: Arbeitsumfang von 1-2 Personen mit max. 1-2 Zeitwochen
  - gerade in Krisenzeiten (Implementierungsphase) braucht man einen Plan (was wäre die Feuerwehr ohne Plan?)

Fertigstellung (Ist) wird dokumentiert durch individuelle Zeiterfassung aufgeschlüsselt in Aktivitäten (*Anette*).

# Anette: Aufgaben-Management

The screenshot displays the Anette software interface, which is used for task management and time tracking. The window title is "Anette" and it includes a "Hilfe" (Help) button. The main menu includes "Management", "Time Browser", "Time Tracker", and "Chat".

The left sidebar shows a hierarchical tree structure:

- ART Bremen
  - Abteilung 2
    - Projekt 1
      - Start
      - Management
      - Ist-Analyse
      - Soll-Analyse
      - Anforderungsspezifikation

The main area is titled "Gantt-Diagramm" and contains a list of tasks with corresponding Gantt bars. The tasks are:

- Start
- Management
- Präsentation Projektplan
- Ist-Analyse
- Vorbereitung Kunden- und Nutzergespräch
- Gespräch mit Kunde
- Gespräch mit Benutzer
- Dokumentation Ist-Zustand und Glossar
- Persona
- Soll-Analyse
- Einarbeitung HIG
- Bestimmung der Anforderungen
- Erstellung von Papier-Prototypen
- Beschreibung Anwendungsfall Persona 1
- Anwendungsfälle
- Beschreibung Anwendungsfall Persona 3
- Beschreibung Anwendungsfall Persona 2
- Beschreibung Anwendungsfall "Bad Guy"
- Anwendungsfälle überprüfen

The Gantt chart shows the duration of each task, with bars colored in blue and orange. The tasks are arranged in a cascading manner, indicating dependencies. The "Erstellung von Papier-Prototypen" task is highlighted in red text.

At the bottom of the window, there is a status bar that reads "0 Objekte in der Synchronisierungs-Liste. Leerlauf" (0 objects in the synchronization list. Idle).

# Anette: Zeiterfassung

Anette Hilfe

Management Time Browser Time Tracker Chat

Neue Tätigkeit starten: \_\_\_\_\_

Tätigkeit: **Projekt 1 -> Management -> Tätigkeit 2**

**diese Tätigkeit starten**

Offene Tätigkeiten: \_\_\_\_\_

ID	Name	Status	Gestartet	Abgelaufene Zeit (Stu...)
1	Tätigkeit 1	Pausiert	11.07.08 13:27	00:01:14
2	Tätigkeit 2	Läuft	11.07.08 13:29	00:00:07

**Tätigkeit wiederaufnehmen**  **Tätigkeit stoppen**

Laufende Tätigkeit: \_\_\_\_\_

**Tätigkeit: Tätigkeit 2**

**Gestartet: Fri Jul 11 13:29:21 CEST 2008**

**abgelaufene Zeit: 00:00:07**

**pause**  **stopp**

Online | 0 Objekte in der Synchronisierungs-Liste. Leerlauf

# Die Aufgabe

Aktuelle Probleme bei der Durchführung von Arbeitssitzungen im SWP und anderswo:

- Terminfindung ist ein langwieriger Prozess
- keine Möglichkeit zur Vorbereitung, da keine Agenda
- schlechte Zeitausnutzung
- wichtige Dinge bleiben unbesprochen
- die Inhalte der Arbeitssitzungen gehen verloren
- Festlegungen in Arbeitssitzungen bleiben unverbindlich
- Aufgaben werden nicht erledigt

Aufgabe: *Meeting Assistant*

- Erstellung eines Systems zur Unterstützung von **Arbeitssitzungen** z.B. im SWP

Rollen: Moderator, Protokollant, sonstige Teilnehmer

Vorbereitung:

- ① Terminfindung
- ② Aufstellung der Agenda mit Priorisierung und Zeitplan
- ③ Versenden der Einladung mit Agenda

## Durchführung:

- ① Feststellung der Anwesenheit
- ② ggf. Anpassung der Agenda
- ③ Abfragen der offenen Aufgaben des letzten Treffens
- ④ Abarbeitung der Agenda:
  - entlang der Priorisierung
  - Einhaltung des Zeitplans
  - Protokollierung aller wesentlichen Punkte
- ⑤ Festlegung von Aufgaben mit Verantwortlichen und Erledigungsdatum

Nachbearbeitung:

- ① Versenden des Protokolls
- ② ggf. Anpassung des Protokolls
- ③ Auswertungen von Statistiken, um Arbeitssitzungen kontinuierlich zu verbessern

*Meeting Assistant* soll den gesamten Prozess unterstützen:

- Zeitfindung unter den Teammitglieder
- Vorbereitung der Agenda und Verteilung im Vorfeld
- Protokollerstellung während der Besprechung und Verteilung im Anschluss
- Überprüfen der offenen Punkte vom letzten Protokoll

Integration mit *Anette* wäre vorteilhaft, ist aber nicht verbindlich.  
Quellcode von *Anette* kann erweitert werden.

## Randbedingungen:

- Implementierung in Java 5 oder höher
- Swing-GUI, keine Webanwendung
- Mehrbenutzerbetrieb
- hoher Anspruch an Datenschutz und -sicherheit
- hoher Anspruch an Benutzbarkeit/Ergonomie
- Client/Server Anwendung
- Verwendung von MySQL für den Server
- Verwendung von SQL-Abfragen
- keine „große“ Datenbank auf dem Clienten (eingebettete sind OK)

## Themen:

- Swing
  - Sockets (TCP)
  - RMI
  - SSL
  - Serialisierung
  - Datenbankbindung
  - OR-Mapping
  - XML Parser
  - Multithreading
  - ...
- 
- jeweils Einführung + betreute Übung
  - optionales Angebot, Teilnahme freiwillig
  - etwa 4–5 Termine
  - Termin: im Januar/Februar?





# Die Beteiligten



	<b>Tel.</b>	<b>E-Mail</b>	<b>Ort</b>
Rainer Koschke	9671	koschke	OAS 3016
Raimar Falke	2576	rfalke	OAS 3012
Markus Eich	64105	eich	DFKI 212
Hendrik Iben	2447	hiben	TAB 1.91
Hui Shi	64260	shi	Cartesium 1.053
Xin Xing	3035	xing	TAB 1.77

... und natürlich **Sie!**


- Web-Seite zur Vorlesung:  
<http://www.informatik.uni-bremen.de/st/Lehre/swp/>
- Folien mit Kommentaren:  
[http://www.informatik.uni-bremen.de/st/lehredetails.php?id=&lehre\\_id=46](http://www.informatik.uni-bremen.de/st/lehredetails.php?id=&lehre_id=46)
- Folien mit Annotationen aus der Vorlesung bei Stud.IP:  
<http://elearning.uni-bremen.de/>
- Video-Aufzeichnung vom letzten Jahr:  
<http://mlecture.uni-bremen.de>
- Newsgroup fb3.lv.swp

## Allgemeine Literatur zur Softwaretechnik:

-  Balzert (1998): Umfassendes deutsches Lehrbuch vmtl. der Bestseller in Deutschland; leider nicht mehr im Buchhandel verfügbar, wird neu aufgelegt
-  Ludewig und Lichter (2006): Umfassendes Lehrbuch, das aus Ludewigs Vorlesungen rund um Softwaretechnik entstanden ist. Diese Vorlesung basiert in großen Teilen auf dem Skript von Ludewigs Vorlesung.
-  Sommerville (2004): Ein Standardlehrbuch, sowohl in deutscher als auch englischer Sprache verfügbar. Im Umfang vergleichbar mit dem Buch von Pressman (2003).
-  Pressman (2003): Ein umfassendes englisches Lehrbuch, das man fast schon als Enzyklopädie bezeichnen könnte. Behandelt auch nicht-objektorientierte Konzepte.

-  Brügge und Dutoit (2004): Eine Einführung in Deutsch mit Schwerpunkt Objektorientierung und UML.
-  Zuser u. a. (2004): Eine Einführung in Deutsch mit Schwerpunkt Objektorientierung und dem Rational Unified Process. Weniger umfassend als das Buch von Brügge und Dutoit (2004).

## Spezialthemen:

-  Störrle (2005): Eine kurze Einführung in die Konzepte der UML 2.0 in deutscher Sprache.

- 1 **Balzert 1998** BALZERT, Helmut: Lehrbuch der Software-Technik. Spektrum Akademischer Verlag, 1998. – derzeit nicht mehr verfügbar, wird neu aufgelegt. – ISBN 978-3827403018
- 2 **Brügge und Dutoit 2004** BRÜGGE, Bernd ; DUTOIT, Allen H.: Objektorientierte Softwaretechnik. Prentice Hall, 2004
- 3 **Ludewig 2003** LUDEWIG, Jochen: Einführung in die Softwaretechnik. Vorlesungs-Skriptum. 2003
- 4 **Ludewig und Lichter 2006** LUDEWIG, Jochen ; LICHTER, Horst: Software Engineering – Grundlagen, Menschen, Prozesse, Techniken. dpunkt.verlag, 2006
- 5 **Maaß 2001** MAASS, Susanne: Soziotechnische Systeme. Vorlesungs-Skriptum. 2001
- 6 **Pressman 2003** PRESSMAN, Roger: Software Engineering – A Practitioner's Approach. Fünfte Ausgabe. McGraw-Hill, 2003

- 7 **Rödiger 2004** RÖDIGER, Karl-Heinz: Vortrag  
Rechtlicher Rahmen der Software-Entwicklung in der Vorlesung  
Software-Projekt. Vorlesungs-Skriptum. 2004
- 8 **Sommerville 2004** SOMMERVILLE, Ian: Software Engineering. Siebte  
Ausgabe. Addison-Wesley, 2004
- 9 **Störrle 2005** STÖRRLE, Harald: UML 2 für Studenten. Pearson  
Studium, 2005. – ISBN 3-8273-7143-0
- 0 **Winter 2003** WINTER, Andreas: Einführung in die Softwaretechnik.  
Vorlesungs-Skriptum. 2003
- 1 **Zuser u. a. 2004** ZUSER, W. ; GRECHENIG, T. ; KÖHLE, M.: Software  
Engineering mit UML und dem Unified Process. Zweite Ausgabe.  
Pearson Studium, 2004