

Software-Projekt

Prof. Dr. Rainer Koschke

Fachbereich Mathematik und Informatik
Arbeitsgruppe Softwaretechnik
Universität Bremen

Wintersemester 2007/08

Überblick I

- 1 Vorbemerkungen
- 2 Softwaretechnik
- 3 Planung

- ① Vorbemerkungen
 - Ziele und Inhalt
 - Angestrebte Resultate
 - Aktivitäten der Software-Entwicklung
 - Ablauf
 - Zeitplan
 - Projektplan
 - Vorstellung der Aufgabe
 - Kontaktdaten
 - Ressourcen
 - Anmeldung
 - 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 Softwaretechnik¹
- Vermittlung von spezifischen Kenntnissen für die Entwicklung des Anwendungssystems X

¹Wird 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 (Commercial off the Shelf)
- 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)

- Softwaretechnik ist nicht nur Programmieren.

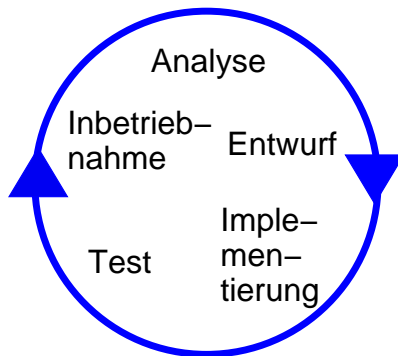
- Softwaretechnik ist nicht nur Programmieren.
- Softwaretechnik ist auch Programmieren.

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

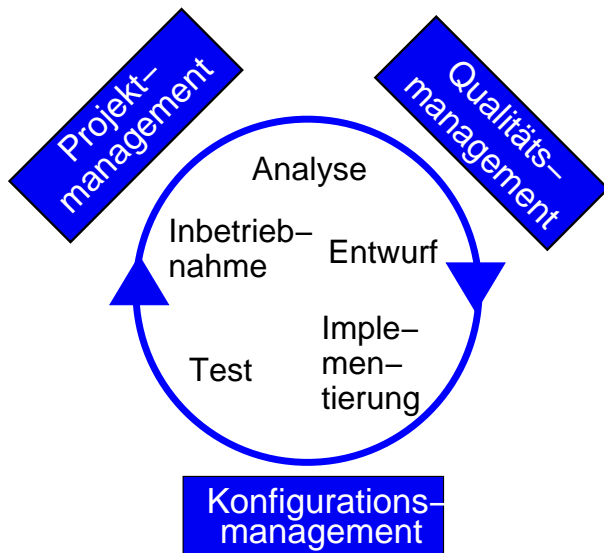
- Softwaretechnik ist nicht nur Programmieren.
- Softwaretechnik ist auch Programmieren.
- Software-Entwicklung produziert Dokumente.
- Der Quellcode ist **ein** Dokument **unter vielen**.

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

Aktivitäten bei der Softwareentwicklung



Aktivitäten bei der Softwareentwicklung



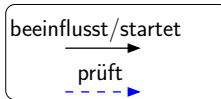
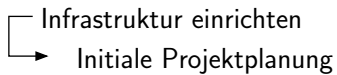
Infrastruktur einrichten

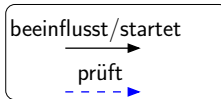
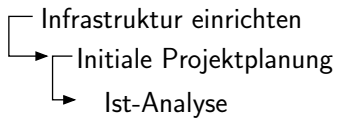
beeinflusst/ startet

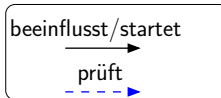
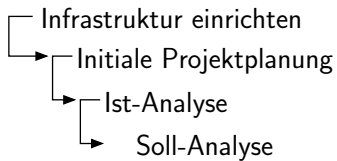


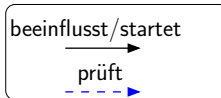
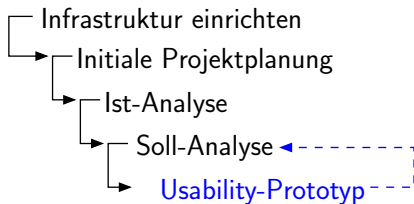
prüft

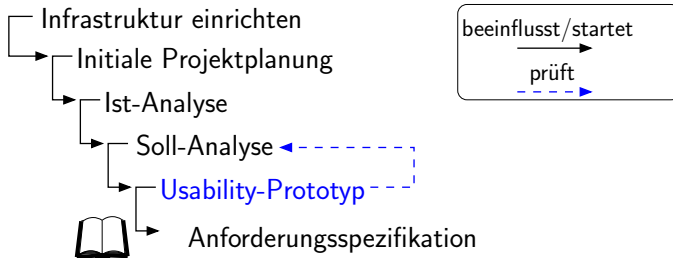


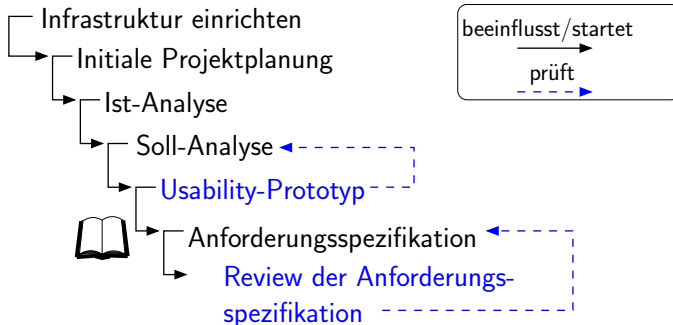


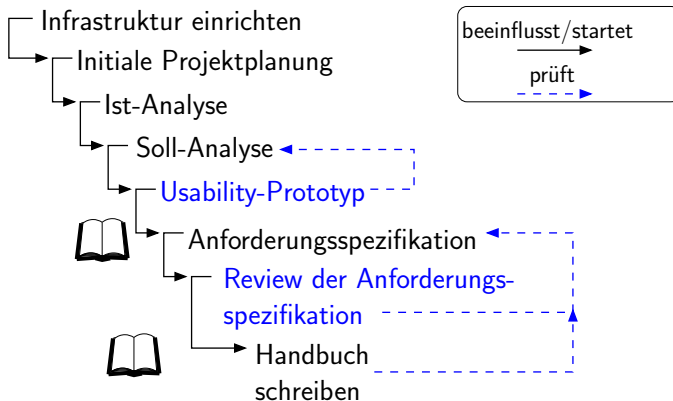


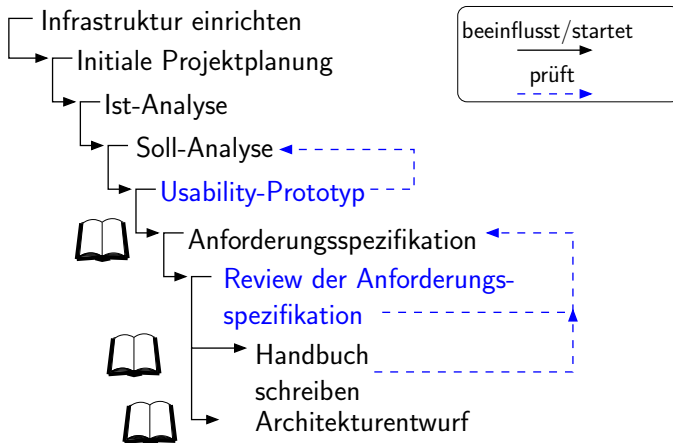


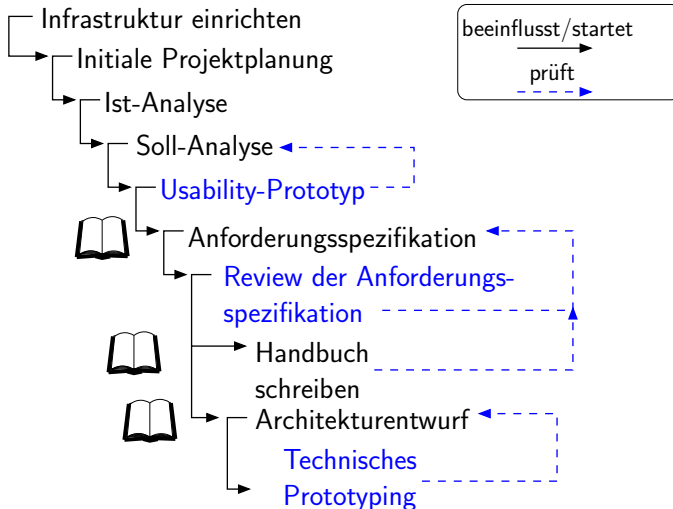


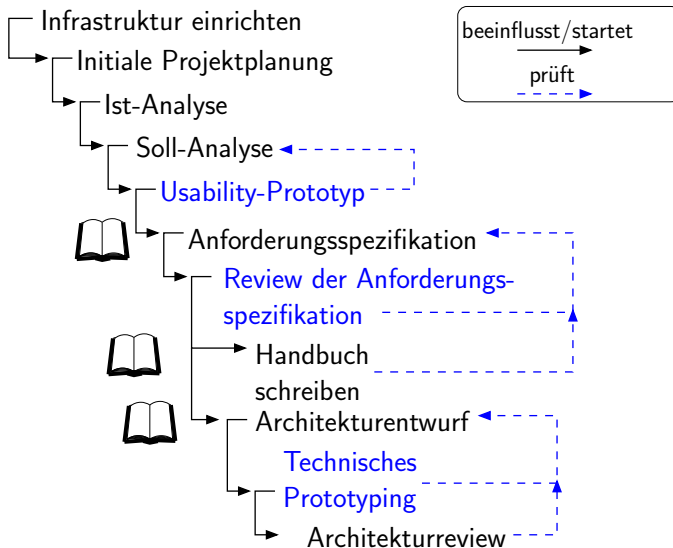


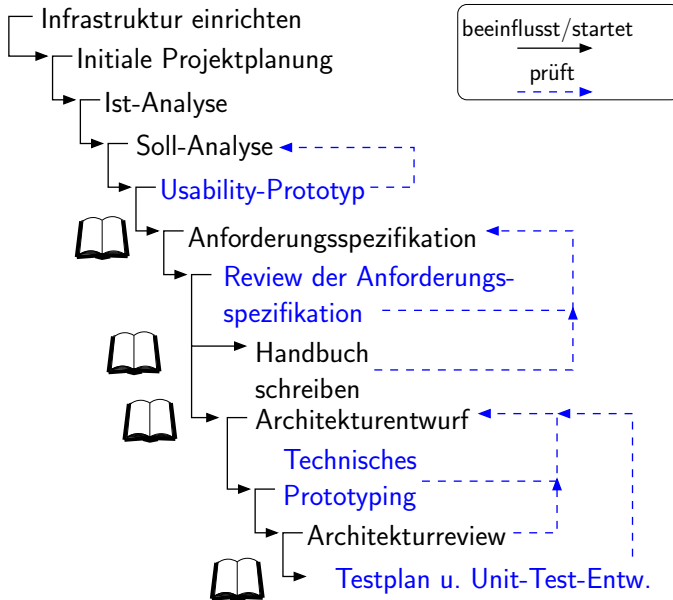


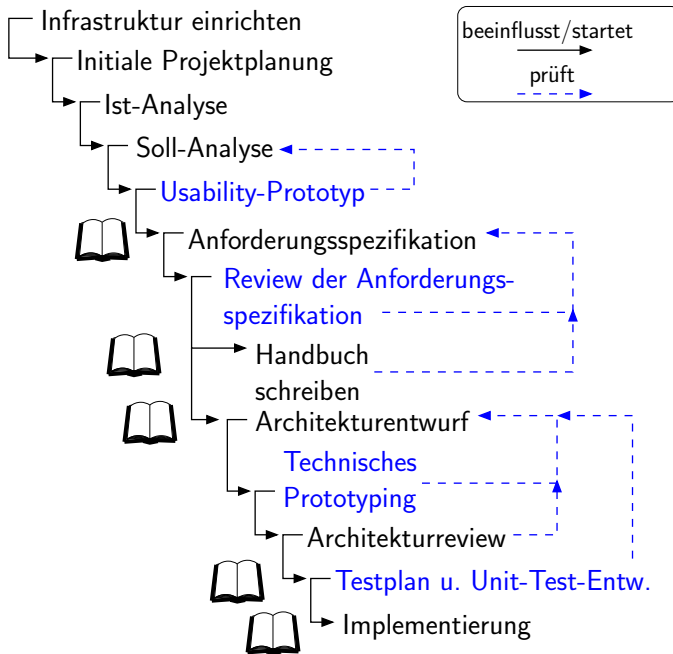


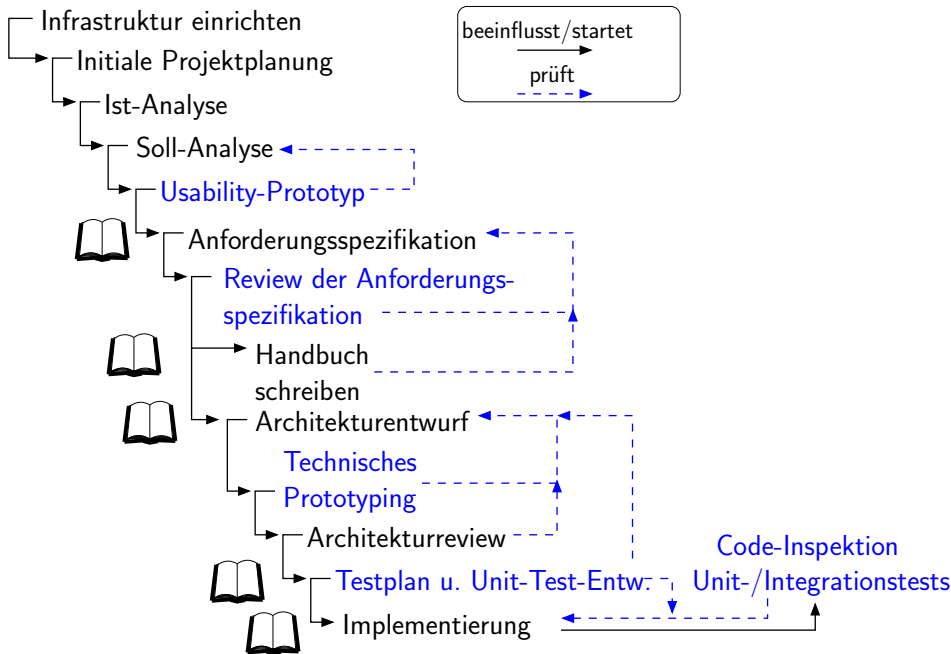


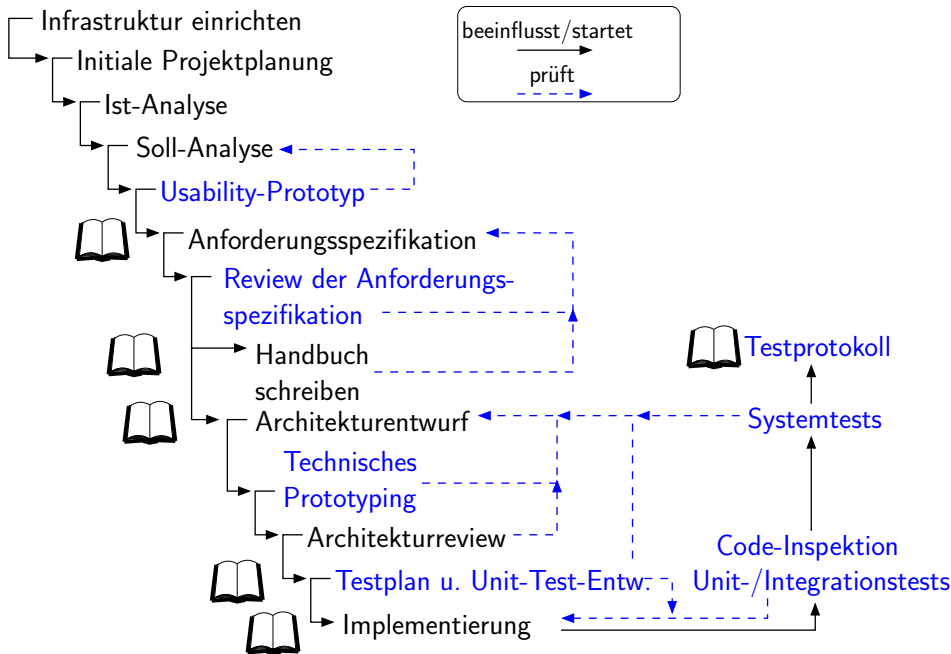


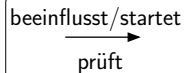


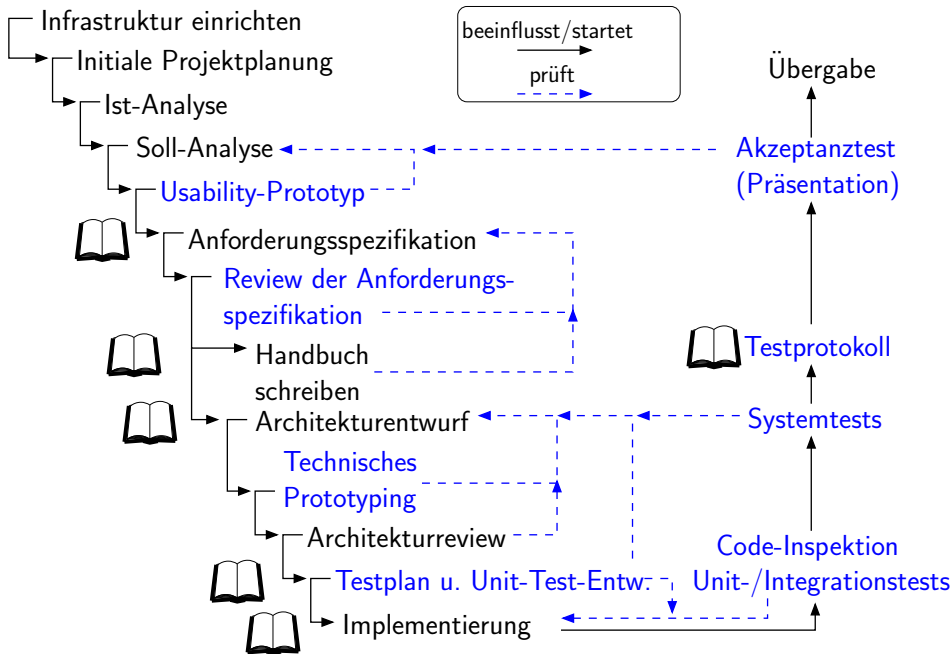












Zeitplan

26.11.2007	Initialer Projektplan
14.1.2008	Anforderungsspezifikation mit Angebot
21.-28.1.2008	Review der Anforderungsspezifikation
11.-14.2.2008	Vorlesung <i>Datenbankgrundlagen</i>
Feb./März 2008	Prüfung zur Vorlesung
20.3.2008	Architekturentwurf
7.-21.4.2008	Walkthrough des Architekturentwurfs
21.4.2008	Testplan mit Unit-Tests
Mai/Juni 2008	Code-Inspektion
11.7.2008	Endabgabe ²
Juli 2008	Abschlusspräsentationen

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

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

Themen:

- Swing
 - Sockets (TCP)
 - RMI
 - SSL
 - Serialisierung
 - Datenbankbindung
 - OR-Mapping
 - XML Parser
 - Multithreading
 - ...
-
- jeweils Einführung + betreute Übung
 - Angebot, Teilnahme freiwillig
 - Termin: Dienstags in E0 – wann?

- initialer Projektplan umfasst Aufgaben, Teilnehmer, Rollen, Verantwortlichkeiten und Pflichten, Risiken und deren Behandlung

Abgaben des inkrementellen Projektplans

- initialer Projektplan umfasst Aufgaben, Teilnehmer, Rollen, Verantwortlichkeiten und Pflichten, Risiken und deren Behandlung
- danach: inkrementelle Fortführung
 - Abgabedatum eine Woche nach Abgabe der abgeschlossenen Phase
 - Verantwortlicher ist 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
 - zu Beginn des Sommersemesters:
ein Plan für die verbleibenden Arbeiten

Abgaben des inkrementellen Projektplans

- initialer Projektplan umfasst Aufgaben, Teilnehmer, Rollen, Verantwortlichkeiten und Pflichten, Risiken und deren Behandlung
- danach: inkrementelle Fortführung
 - Abgabedatum eine Woche nach Abgabe der abgeschlossenen Phase
 - Verantwortlicher ist 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
 - zu Beginn des Sommersemesters:
 - ein Plan für die verbleibenden Arbeiten

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

Aufgabe:

- Erstellung eines Systems zur **Arbeitszeiterfassung**

Wozu?

- Projektcontrolling
 - Bestandteil des Projektmanagements
 - Soll-/Ist-Vergleich
 - Überschreitung des Budgets?
- Verbesserung zukünftiger Schätzungen
 - Erfahrungsdatenbank
- Eigenes Zeitmanagement
 - Wo ist die Zeit geblieben?

- Definition der Arbeitspakete
- Aufwandsschätzung
- Einplanung der verfügbaren Ressourcen
- Verfolgung des Projektfortschritts → erledigte APs
- Frühe Erkennung von Planabweichungen
- evtl. Anpassung der Planung

WR Time Tracker SWP

[Logout](#) | [Profil editieren](#) | [Anmeldung zur Präsentation](#) | [Hilfe](#)

[Meine Zeiten](#) | [Berichte](#) | [Projekte](#) | [Aktivitäten](#) | [Personen](#)

Jochen Quante (Tutor), ITD [1]

Meine Zeiten: Mittwoch, 24. Oktober 2007

Datum: Mittwoch, 24. Oktober 2007

Projekt:

Aktivität:

Unteraktivität:

Start: (hh:mm)

Ende: (hh:mm)

Dauer: (hh:mm)

<<< Oktober 2007 >>>						
Mon	Die	Mit	Don	Fre	Sam	Son
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
<<		Heute			>>	

Beschreibung:

WR Time Tracker SWP

[Logout](#) | [Profil editieren](#) | [Anmeldung zur Präsentation](#) | [Hilfe](#)

[Meine Zeiten](#) | [Berichte](#) | [Projekte](#) | [Aktivitäten](#) | [Personen](#)

Jochen Quante (Tutor), JQ-Test [99]

Aktivitäten

Aktivitäten			
Name	Projekt	Editieren	Löschen
Anforderungsspez.	--- all ---	Editieren	Löschen
Architektur	--- all ---	Editieren	Löschen
Auslieferung	--- all ---	Editieren	Löschen
Dokumentation	--- all ---	Editieren	Löschen
Implementierung	--- all ---	Editieren	Löschen
Projektplanung/Mgmt.	--- all ---	Editieren	Löschen
- Risikomanagement	--- all ---	Editieren	Löschen
Sonstiges	--- all ---	Editieren	Löschen
Teambesprechung	--- all ---	Editieren	Löschen
Test	--- all ---	Editieren	Löschen
Tutorium	--- all ---	Editieren	Löschen

[Neue Aktivität hinzufügen](#)

WR Time Tracker SWP

[Logout](#) | [Profil editieren](#) | [Anmeldung zur Präsentation](#) | [Hilfe](#)

[Meine Zeiten](#) | [Berichte](#) | [Projekte](#) | [Aktivitäten](#) | [Personen](#)

Jochen Quante (Tutor), JQ-Test [99]

Berichte - Stundenzettel

Benutzer wählen

- ☒ Jochen Quante
☐ Jochen Quante (Test)

Projekt wählen

SWP

Aktivität wählen

--- all ---

Zeitraum wählen

Planungsphase

oder Daten eingeben

Anfangsdatum: 01.10.2007 (Tag.Monat.Jahr)

Enddatum: 31.10.2007 (Tag.Monat.Jahr)

Felder anzeigen

Projekt: <input type="checkbox"/>	Beschreibung: <input checked="" type="checkbox"/>	Start: <input checked="" type="checkbox"/>	Tage ohne Eintrag: <input type="checkbox"/>
Aktivität: <input checked="" type="checkbox"/>	Unteraktivität: <input type="checkbox"/>	Ende: <input checked="" type="checkbox"/>	Dauer: <input checked="" type="checkbox"/>

Gruppieren nach

Benutzer

Bericht erstellen

- Internetzugang erforderlich
- Einloggen auf Webseite erforderlich
- Timeout
- Umständlich → wird gern vergessen
- Arbeitspakete sind nur im Projektplan definiert, Zuordnung schwierig
- Taugt kaum zur Fortschrittskontrolle

Erforderliche Funktionalität

- Verwaltung von hierarchischen Arbeitspaketen (mind. eindeutige ID, Name, Beschreibung, Verantwortlicher, Bearbeiter, zeitliche Rahmenbedingungen, geplante Ressourcen, Fortschritt)
- Verwaltung von Ressourcen, insbes. Arbeitszeit
- Verwaltung von Teammitgliedern
- einfache Arbeitszeiterfassung mit Start, Stop, Wahl des Arbeitspakets, Eingabe einer Tätigkeitsbeschreibung
- Übersicht über alle Arbeitspakete und deren Stand
- zentrale Ablage aller Daten in einer MySQL-Datenbank
- Online- und Offlinebetrieb
- Export der Arbeitszeiten zum Timetracker
- zusätzlich für Gruppengröße ab 6: Import von Daten aus ganttproject
- zusätzlich für Gruppengröße 7: Statistiken (Charts)

Randbedingungen:

- Implementierung in Java 5
- Verwendung von MySQL
- Verwendung von SQL-Abfragen
- Swing-GUI
- Mehrbenutzerbetrieb
- hoher Anspruch an Datenschutz und -sicherheit
- hoher Anspruch an Benutzbarkeit/Ergonomie
- Client/Server Anwendung

Nächste Woche!

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

Die Beteiligten

	Tel.	E-Mail	Ort
Rainer Koschke	9671	koschke	OAS, Linzer Str. 9a
Raimar Falke	2576	rfalke	OAS, Linzer Str. 9a
Pierre Frenzel	2576	saint	OAS, Linzer Str. 9a
Mirko Horstmann	7835	mir	TAB 2.78
Jochen Quante	2421	quante	OAS, Linzer Str.9a
Jörn Sprado	4781	sprado	TAB 2.58

... und natürlich **Sie!**

- Web-Seite zur Vorlesung:
<http://www.informatik.uni-bremen.de/st/Lehre/swp/>
- Folien:
http://www.informatik.uni-bremen.de/st/lehredetails.php?id=&lehre_id=45
- Bei Stud.IP anmelden - annotierte Folien:
<http://elearning.uni-bremen.de/>
- Video-Aufzeichnung:
<http://mlecture.uni-bremen.de>
- Newsgroup fb3.lv.swp





Ab sofort bis zum 1.11.2007, 23.59 Uhr.

Web-Seite zur Anmeldung:

<http://www.informatik.uni-bremen.de/st/mems/>

- gruppenweise anmelden!
- einen Gruppennamen sorgfältig auswählen
- ausländische Studierende integrieren

Allgemeine Literatur zur Softwaretechnik (absteigend sortiert in Bezug auf den Umfang):

-  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.

2 Softwaretechnik

- Eigenschaften von Software
- Software-Lebenszyklus
- Software-Evolution
- Entstehung der Softwaretechnik
- Merkmale der Softwaretechnik

Was ist Software?

Definition

Software: Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.
IEEE Std 610.12-1990 (1990)

Ein Produkt wie jedes andere?

Software

- ist ein technisches Produkt,
 - weist aber einige einmalige Merkmale auf.
- Sie sollten beim Umgang mit Software beachtet werden.

- Software ist immateriell:
 - Software kann nicht ohne Weiteres betrachtet werden.
 - Kopie und Original sind völlig gleich.
 - Software wird nicht gefertigt, sondern nur entwickelt.
 - Software verschleißt nicht.
 - Software „altert“ in dem Maße, in dem sich ihre Umwelt ändert.

- Software hat keine natürliche Lokalität.
- Ihre Werkstoffe sind amorph und universell.
- Strukturen müssen aktiv geschaffen werden.
- Ein Programm realisiert keine stetige Funktion.
- Korrektheit ist durch Test nicht prüfbar.

- Software-Systeme sind sehr komplex:
 - Entwicklungskosten sind unvermeidlich hoch.
 - Korrekte Konstruktion ist extrem schwierig.
- Software spiegelt (in vielen Fällen) die Realität wider.

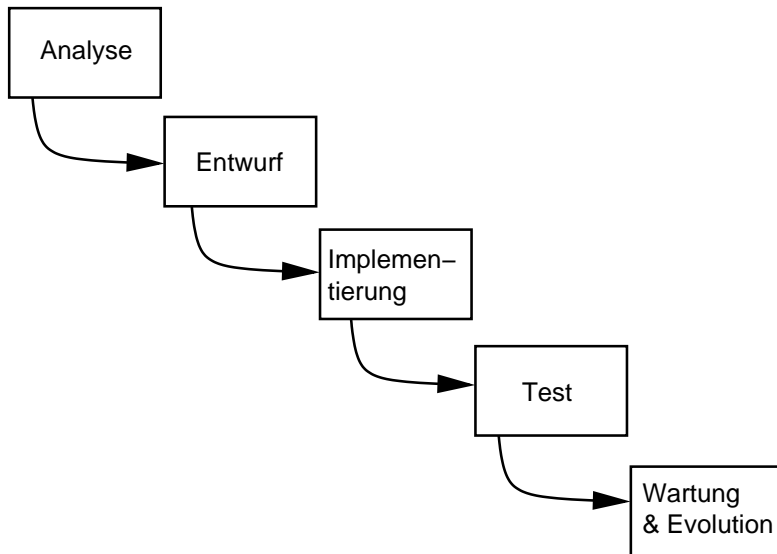
“This complexity is compounded by the necessity to conform to an external environment that is arbitrary, unadaptable, and everchanging.”

–F.P. Brooks, 1987

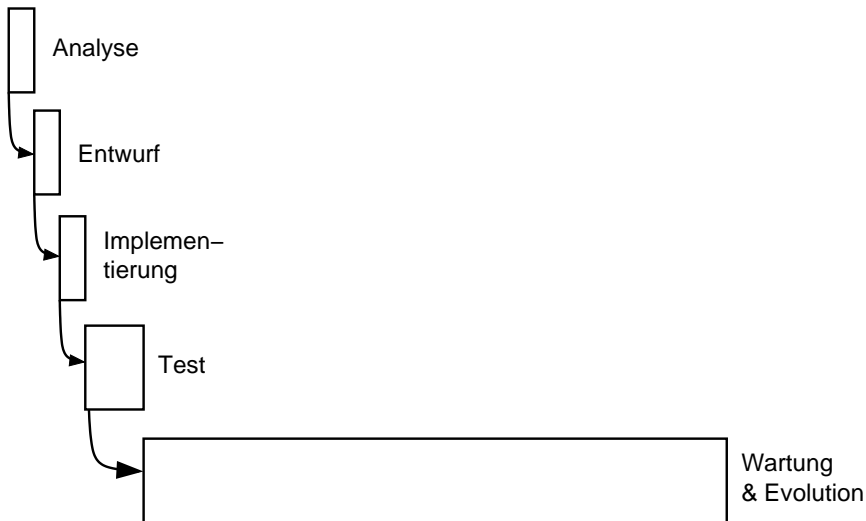
- Software gilt als flexibel (d.h. leicht änderbar) ³:
 - Sie verbindet Hardware und Umgebung.
 - Details der Aufgabenstellung werden in aller Regel durch Software realisiert.
- Änderungen schlagen sich entsprechend primär in der Software nieder.

³Software ist nur in dem Maße flexibel, in dem die Flexibilität explizit eingebaut ist.

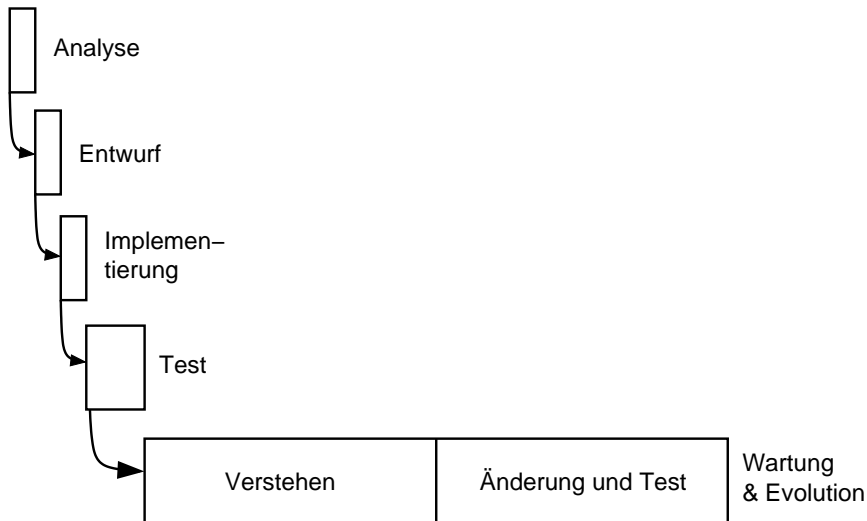
Software-Lebenszyklus



Aufwand in den Phasen nach McKee (1984)



Evolutionsphase nach Fjeldstadt und Hamlen (1984)

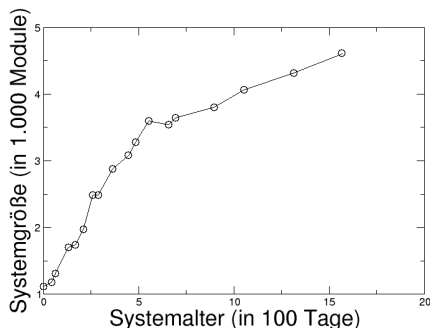


Gesetz vom fortwährenden Wandel

- Software löst ein Problem der realen Welt,
- die reale Welt ändert sich,
- Software muss sich anpassen,
- bis sie abgelöst wird.

Gesetz der zunehmenden Komplexität

- die Komplexität der Software erhöht sich beim Wandel,
- wenn keine Gegenmaßnahmen ergriffen werden



Konsequenzen:

- Änderbarkeit ist eine Schlüsselqualität
- Änderbarkeit muss frühzeitig angestrebt werden
 - sie lässt sich nicht nachträglich überstülpen
- wer sie vernachlässigt, nimmt einen Kredit auf, den er teuer abzahlen wird

Wo liegen die Probleme in der Softwaretechnik?

In den späten 60er Jahren:

- Manche Software-Projekte sind selbst mit gigantischem Aufwand nicht zu schaffen.
- In vielen anderen Fällen wird der Abschluss nur mit unbefriedigenden Resultaten, viel zu spät und/oder mit extremen Kostenüberschreitungen erreicht.
- Andererseits erzielt die Hardware-Entwicklung atemberaubende Fortschritte.

“The whole trouble comes from the fact that there is so much tinkering with software. It is not made in a clean fabrication process, which it should be. What we need is software engineering.”

–F.L. Bauer, 1968

Wie viel hat sich bis heute daran geändert?

Beispiel

- Fehler:
vollständiger Batterieausfall in Fahrzeugen eines deutschen Automobil-Herstellers
- Ursache:
Fehler in der software-gesteuerten Innenbeleuchtung
- Folge:
Rückrufaktion mit einem (geschätzten) Schaden von mehreren Millionen DEM

Partsch (1998)

Wie viel hat sich bis heute daran geändert?

Beispiel

- Fehler:

Bei der rechnergestützten Auswertung der OB-Wahl in Neu-Ulm 1994 wurde zunächst eine Wahlbeteiligung von 104% festgestellt.

- Ursache:

In der Auswertungssoftware hatte sich ein mysteriöser Faktor 2 eingeschlichen.

- Folge:

Neuauszählung

Partsch (1998)

Wie viel hat sich bis heute daran geändert?

Beispiel

- Fehler:
USS Yorktown treibt während eines Manövers im September 1997 manövrierunfähig bei Cape Charles, VA.
- Ursache:
Fehler in einer Routine zum Abfangen einer Division durch Null in einer Windows NT Applikation. Die „Null“ wurde als manuelle Eingabe einer enormen Datenmenge interpretiert.
- Folge:
“Atlantic Fleet officials said the ship was dead in water for about 2 hours and 45 minutes.”

Neumann (1999)