
Software Systems Engineering

Veranstaltungen: Sommersemester 2010

Prof. Dr. Klaus Schmid

Warum ist Software Engineering wichtig?

Software – das Fundament unserer Wirtschaft

Verbreitung in verschiedensten Gebieten:

- Informationssysteme
 - Buchhaltung
 - Finanz- und Abrechnungssysteme
 - Web-Portale
- Telekommunikation
 - Handys
 - Datenübertragung
 - VoIP
- Eingebettete Systeme
 - Consumer-Elektronik
 - Automobil-Industrie
 - Luft- und Raumfahrt



Warum ist Software Engineering wichtig?

Bedeutung der IT-Industrie in Deutschland

Deutschland: ITK (ohne eingebettete Systeme)

- Umsatz 2006: ~290 Milliarden EUR (Zahlen nach BITKOM)
- Stärkste Wachstumsbereiche: Softwareentwicklung und IT-Services
Software alleine: 4,5 – 6 % in den letzten Jahren
- Beschäftigte 800.000 (in 2006)
 - Anstieg der Beschäftigten in Software und Dienstleistung um 20 % (seit 2000)
 - 10.000 neue Arbeitsplätze 2006
 - 50% aller Unternehmen nennen Fachkräftemangel als wesentliches Wachstumshindernis

Warum ist Software Engineering wichtig?

Extreme Kosten- und Zeitüberschreitungen

- 1981: US Air Force Command & Control Software überschreitet Kostenvoranschlag fast um den Faktor 10: **3,2 Mio. US-\$.**
- 1987-1993: Integration der kalifornischen Systeme zur Führerschein- und KFZ-Registrierung abgebrochen: **1,5 Mio. US-\$.**
- 1992: Integration des Reservierungssystems SABRE mit anderen Reservierungssystemen abgebrochen: **55 Mio. US-\$.**
- 1997: Entwicklung des Informationssystems SACSS für den Staat Kalifornien abgebrochen: **500 Mio. US-\$.**
- 1994: Eröffnung des Denver International Airport um 16 Monate verzögert wegen Softwareproblemen im Gepäcktransportsystem: **655 Mio. US-\$.**
- 2003: Das deutsche Maut Erfassungssystem "Toll Collect" konnte nur verspätet in Betrieb genommen werden: **~6,5 Mrd. €.**

Warum ist Software Engineering wichtig?

CHAOS Report

- Jährlicher Bericht seit 1994 über den Erfolg von IT-Projekten
- Es wurden ca. 100.000 IT-Projekte in den USA untersucht
- Herausgeber: Standish Group International, Inc.

CHAOS Report ordnet IT-Projekte in drei Kategorien ein

- **Successful:** Projekt wurde innerhalb der vorgegebenen Zeit und Budget abgeschlossen. Projektergebnis ist im Einsatz und erfüllt alle Anforderungen.
- **Challenged:** Projekt ist abgeschlossen. Projektergebnis ist im Einsatz. Zeit, Budget oder Leistung sind aber nicht im vorgegebenen Umfang.
- **Failed:** Das Projekt wurde vorzeitig abgebrochen oder das Projektergebnis wurde nie eingesetzt.

Warum ist Software Engineering wichtig?

Erfolgswahrscheinlichkeit der Projekte in Abhängigkeit der Jahre

	Succeeded	Failed	Challenged
1994	16%	31%	53%
1996	27%	40%	33%
1998	26%	28%	46%
2000	28%	23%	49%

[Quelle: CHAOS Report, Standish Group International, Inc.]

Warum ist Software Engineering wichtig?

Software Engineering als Industrielle Basis

- Grundlage erfolgreicher Produktentwicklung
- Notwendig interdisziplinär:
 - Organisatorische und Management-Aspekte
 - Technische Aspekte
 - Betriebswirtschaftliche Aspekte

Die Qualität der Softwareproduktion bestimmt die wirtschaftlichen Möglichkeiten von Unternehmen!

Warum ist Software Engineering wichtig?

Katastrophen in der Verwendung

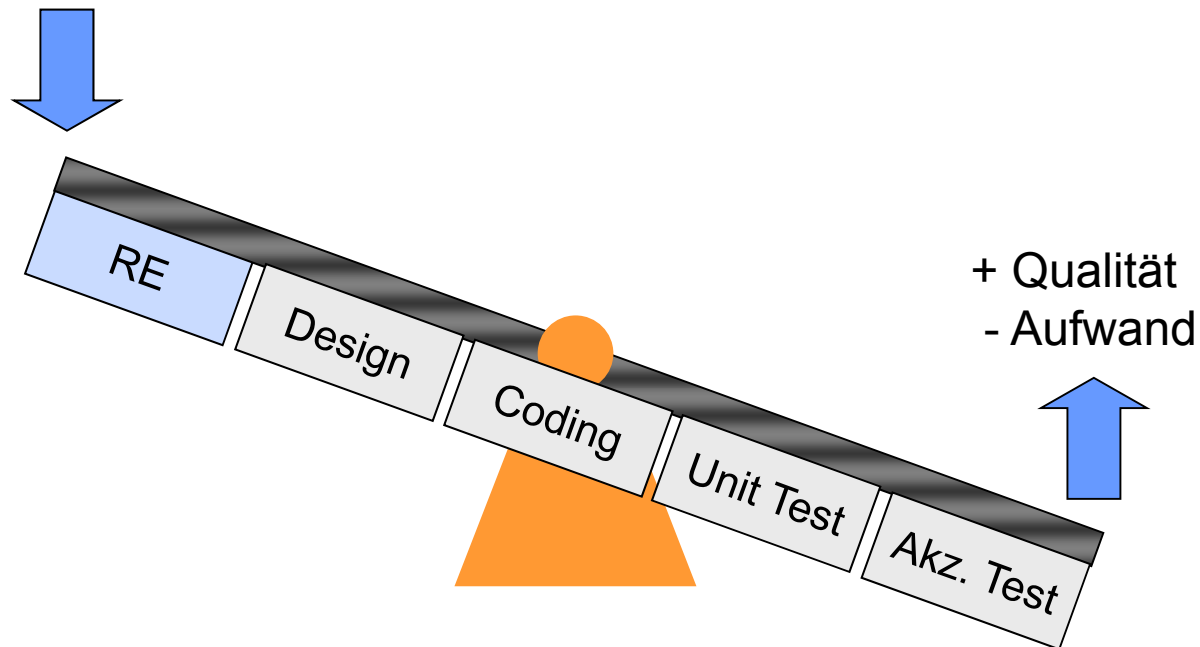
- 1988: Ein Airbus schießt über die Landebahn hinaus, da sich bei Aquaplaning die Schubumkehr nicht einschalten ließ.
- 1999: Verlust der Sonde "Mars Climate Orbiter" wegen falscher Einheitenumrechnung.
- 1999: 20.500 3er BMWs müssen wegen eines Software-Bugs in der Airbag-Steuerung zurückgerufen werden. 50% aller Autopannen sind bereits auf Ausfälle der Bordelektronik zurückzuführen, Tendenz steigend.
- 2002: Aufgrund eines Softwareproblems konnten mit Postbank-EC-Karten bei allen anderen Geldinstituten außer der Postbank selbst mit beliebigen Pincodes Euro abgeboben werden, ohne dass das Sparkonto mit der abgehobenen Summe belastet wurde.
- 2004: Siemens S65 wird wegen Softwarefehlern, die Hörschäden verursachen können, aus dem Handel genommen.



Requirements Engineering

Warum: Requirements Engineering

Besseres RE



Anforderungsfehler, die erst im Test entdeckt werden, sind 100x so teuer wie bei Entdeckung im RE!

Veranstaltungen

Veranstaltungen 1. Semester im Wintersemester 2007 / 08

- Informatik I (erste Veranstaltung: **23.10.2007**)

- Vorlesung dienstags 16-19h H3
- Saalübung dienstags 19-20h H1
- Präsenzübung freitags 8-10 / 10-12h C 213 Spl

- Programmierung I: Java (erste Veranstaltung: **22.10.2007**)

- Vorlesung montags 14-16h C 147 Spl
- Übung montags 16-18h C 147 Spl
- mittwochs 8-10 / 10-12h C 147 Spl

Projektanteil: März

Veranstaltungen

Weitere Veranstaltungen im Wintersemester 2007 / 08

- Grundlagen des Software Engineering (GSE)
(erste Veranstaltung: **24.10.2007**)

– Vorlesung	mittwochs	10-13h	A9 Spl
– Saalübung	mittwochs	13-14h	A9 Spl
– Präsenzübung	donnerstags	12-14h	A9 Spl

- Requirements Engineering (erste Veranstaltung: **24.10.2007**)

– Vorlesung	mittwochs	8-10h	A9 Spl
– Übung	donnerstags	8-10h	C213 Spl

Veranstaltungen

Übersicht der Veranstaltungen

Bachelor

- Grundlagen des Software Engineering

Master

- Softwareproduktlinienentwicklung
- Prozesse und Management des Software Engineering
- Fortgeschrittenenpraktikum: Software Engineering

Veranstaltungen

Bachelor: Grundlagen des Software Engineering

- Entwicklung von Systemen wird schwierig, wenn es in Gruppen stattfindet
- Wie geht man dann vor?
- Was gehört noch dazu außer zu programmieren?
 - Was will der Kunde?
 - Wie entwerfe ich große Systeme?
 - Wie stelle ich sicher, dass das System das tut, was es soll?

Achtung: Diese Veranstaltung ist 4. Semester Pflicht für:

Studenten der Wirtschaftsinformatik & IMIT Neue Prüfungsordnung

Vorlesung Mittwoch 10:15-13:00 A9, Übung Donnerstag 10:15 A9

Veranstaltungen

Bachelor: Grundpraktikum Softwaretechnik (IMIT/WINF)

- Einfache Softwareentwicklungsprozesse für kleine Gruppen
- Geringe Ansprüche an Planung, daher auch ohne SE-Vorkenntnisse möglich
- Technik-orientiert

Bachelor: Grundpraktikum: Systematische Entwicklung von Informationssysteme (nur WINF)

- Basiert auf gleichen Grundlagen wie Softwaretechnikpraktikum
- Spezieller Fokus auf der Entwicklung von typischen Systemen der Wirtschaftsinformatik

Achtung: Wegen Überlappung kann nur eine der Veranstaltungen belegt werden

Gemeinsame Vorbesprechung Donnerstag: 16.4.09, 14:15 , C147

Master: Software Produktlinien

- Ein neues Programm soll erstellt werden.
 - Das habe ich doch schon mal programmiert, oder?
 - Alter Code passt aber nicht ganz! — Wird passend gemacht!
 - Habe einen Fehler gefunden... muss überall nachgezogen werden!
- Firmen haben ähnliche Probleme
 - Anforderungen, Architektur, Code wird kopiert
 - Riesiges Potential bei richtiger Systematik!
 - Betrifft nicht nur Programmierer sondern ganzes Unternehmen!

Vorlesung Dienstag 13:15 -15:45 P1a, Übung Mittwoch 12.00 P1a

Veranstaltungen

Master: Prozesse und Management des Software Engineering

- Fokussiert die Planung und den Ablauf von Softwareentwicklung:
 - Prozesse beschreiben: **Dokumentation / Modellierung**
 - Projektziele formulieren und Erfüllung überprüfen: **Messen**
 - Qualität vorhersagen können: **Abläufe standardisieren**
 - Wie standardisiert sind die Abläufe in einer Organisation: **Reifegrade?**
- Welche Unterstützungstätigkeiten gehören zu einer strukturierten Entwicklung von Software?
 - Aufgaben- / Rollenverteilung
 - Management

Vorlesung Dienstag 16:15-17:45 A9, Übung Donnerstag 14-16 Uhr C213

Veranstaltungen

Information:

- Bei Interesse am Thema:
 - Projektarbeiten
 - Bachelorarbeiten
 - Masterarbeiten

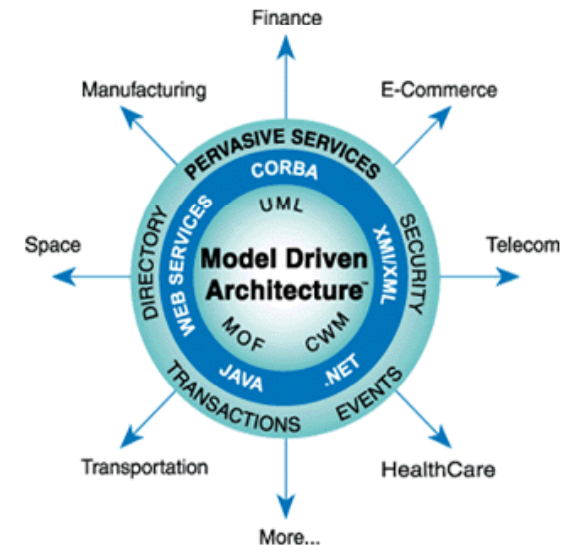
 - Stellen als wissenschaftliche Hilfskraft in der Forschung
- Stellen als wissenschaftliche Hilfskraft in der Lehre:
 - Dringend: bereits bei Interesse für nächstes Semester (WiSe) melden!



Veranstaltungen

Modellbasierte Software-Entwicklung

- UML?
 - Da malt man doch nur Bilder
 - Schreibt die Bilder als Programm ab
 - Wirft die Bilder weg...
- Modelle sind zukunftssicher
 - Dokumentation
 - Ziel: Möglichst viel Generieren um typische Fehler zu minimieren
- Aber: UML ist nicht alles
 - Frühe Modelle z.B. für Anforderungen
 - Werkzeugketten
 - Produktionsumgebungen für Produktlinien



Vorlesung Mittwoch 12.00 C213, Übung Donnerstag 12.00 C213

Seminar: Flexible Benutzerschnittstellen systematisch entwickeln

- Die Benutzerschnittstelle ist das „Gesicht“ jedes IT-Systems
 - Gute Benutzerschnittstelle \Rightarrow es macht Spaß
 - Schlechte Benutzerschnittstelle \Rightarrow es nervt, dauert, man macht Fehler
- Die Entwicklung guter Benutzerschnittstellen wird häufig als Kunst statt als Wissenschaft empfunden
- Benutzerschnittstellen sollten flexibel sein und sich an die Benutzerbedürfnisse anpassen
- Welche Methoden können genutzt werden zur systematischen Entwicklung von Benutzerschnittstellen?

Beginn: 17.4.08, Seminarvorbesprechung: 10:15 C213
Anmeldung im Moodle

Veranstaltungen

Weitere Veranstaltungen im Wintersemester 2007 / 08

Wesentlicher Ansatz zur Effizienzsteigerung

- Drastische Reduktion des Entwicklungsaufwands (teilweise Faktor 10)
- Deutlich höhere Qualität der Systeme (50% Defect Density)
- Deutliche Reduktion der Wartungskosten (60%)

Bosch, Nokia, Siemens,

Seminar: Softwareproduktlinien

Vorbesprechung Donnerstag 25.10 10-12h B26 Spl

Anmeldung über Moodle:

Themenanfragen direkt per Email an stender@sse.uni-hildesheim.de