

**Software Engineering**  
*The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*  
IEEE Std. 610.12 (1990)

## Methoden des Software Engineering

Nora Koch, Harald Störrle, Martin Wirsing

in Zusammenarbeit mit  
Florian Hacklinger

WS 2004/05

Block R (Rahmen): Einführung

## 4. Juni 1996: Erster Start der "Ariane-5"



- Während des Flugs läuft ein unnötiges Kalibrierungsprogramm für die Trägheitssensoren. Die gemessenen Werte der Ariane-5 überschreiten die in der Ariane-4-Software vorgesehenen Bereiche.
- Die dadurch ausgelöste (Ada-)Exception wird durch Anhalten des Steuerungscomputers behandelt, um auf ein zweites redundantes System umzuschalten.
- Im zweiten System tritt der gleiche Software-Fehler auf und wird identisch behandelt.
- Kosten des Ariane-5-Programms bis 1996 ca. 7 Milliarden US-\$
- Wert des zerstörten Satelliten: ca. 500 Millionen US-\$

Methoden des Software Engineering WS04/05, N. Koch, H. Störrle, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Informationsveranstaltung

Einladung an alle Studierenden

(zu Beginn) des Hauptstudiums Informatik

am

28. Oktober 2003, 16.15 Uhr, Hörsaal 1.14, Oettingenstraße

Themen:

- Regelungen zum Hauptstudium
- Fachliche Schwerpunkte am Institut
- Praxisbericht

PST- Informationen: 18. November, 14 Uhr c.t., Raum E 03

Methoden des Software Engineering WS04/05, N. Koch, H. Störrle, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Software-Katastrophe: Kein Einzelfall

- Technik-Katastrophen:**
  - September 1999: Verlust der Sonde "Mars Climate Orbiter" wegen falscher Einheitenrechnung
  - 1985-1987 Therac 25 (Strahlengerät zur Krebsbehandlung): Fehlerhafte Programmierung führt zu Verbrennungen und Todesfällen
- Finanzielle Katastrophen:**
  - 1990 AT&T Telefonverbindung zwischen Ost- und Westküste der USA wg eines SW-Fehlers für mehr als 24 Std unterbrochen: ca. 1 Mia US-\$
  - 1992: Integration des Reservierungssystems SABRE mit anderen Reservierungssystemen abgebrochen: 165 Mio. US-\$
  - 1996 Deutsche Telekom: "1. Januar 1996 ist kein Feiertag": >50 Mio DM
- Terminkatastrophen:**
  - 1994: Eröffnung des Denver International Airport um 9 Monate verzögert wegen Softwareproblemen im Gepäcktransport-System
  - 2003: Einführung des LKW-Mautsystems in Deutschland verzögert sich um 18 Monate

Methoden des Software Engineering WS04/05, N. Koch, H. Störrle, M. Wirsing, LMU München

Vorlesung: Methoden des Software Engineering  
Block R (Rahmen)

## Einführung und Übersicht

Martin Wirsing

Einheit R1, 19.10.2004

Block R (Rahmen): Einführung

## Permanente Software-Krise?

- 1965: Der Begriff der Softwarekrise etabliert sich in Industrie und Wissenschaft.**
  - Fehler in Computersystemen sind fast immer auf Softwarefehler zurückzuführen
  - Software wird nicht termingerecht und/oder zu höheren Kosten als geschätzt fertiggestellt
  - Software entspricht oft nicht den Anforderungen ihrer Benutzer
- Studie von 1979 zu Softwareprojekten (USA):**
  - 75% der Ergebnisse nie eingesetzt
  - 19% der Ergebnisse stark überarbeitet
  - 6% benutzbar.
- Studie von 1994 zu Software-Großprojekten (IBM Consulting):**
  - 55% Kostenüberschreitung
  - 68% Terminüberschreitung
  - 88% Bedarf für starke Überarbeitung

Methoden des Software Engineering WS04/05, N. Koch, H. Störrle, M. Wirsing, LMU München

## Was ist Software?

- **Software ist eine Sammlung von Computerprogrammen, Prozeduren, Regeln, zugehöriger Dokumentation und Daten [IEEE Std. 729-83]**
- **Software ist eine Menge von Programmen oder Daten zusammen mit begleitenden Dokumenten, die für ihre Anwendung notwendig oder hilfreich sind [Hesse et al. 1984]**
- **Verschiedene Sichten auf Software:**
  - Innensicht: Software als System
  - Außensicht: Software als Produkt

## Besonderheiten von Software

- Software ist **immateriell**.
- Software unterliegt **keinem Verschleiß**.
- Es gibt **keine Software-Ersatzteile**:  
Defekte sind immer Konstruktionsfehler.
- Software ist **schwer zu vermessen** („Technische Daten“ von Software?).
- Software gilt als relativ **leicht änderbar**  
(im Vergleich zu materiellen technischen Produkten).
- Software unterliegt einem **ständigen Anpassungsdruck**.
- Software **altert**.

*Software is a hybrid, halfway between an abstract idea and a physical, tangible thing. Software is neither land nor sea, but swamp: a hybrid too thin for the army (software engineering) and too thick for the navy (computer science).  
Brad Cox*

## Innensicht der Entwickler: Software als System

- **System**  
Menge von Elementen, zwischen denen bestimmte Beziehungen bestehen [Duden 1974]
- **Softwaresystem**
  - Basisssoftware: Schnittstelle zur Hardware (insbes. Betriebssysteme wie Unix, Windows, ...), Datenbanken, Kommunikationsprotokolle
  - Middleware: Plattform insbesondere für verteilte Anwendungen (Workflow-Management, CORBA, Grid-Plattform,...)
  - Anwendungssoftware zur Lösung kundenspezifischer Aufgaben
- **Software-intensives System**  
System, das Softwarekomponenten enthält und in dem Software eine große Rolle spielt (eingebettete Systeme in Zügen, Autos; verteilte Anwendungen wie virtuelle Firmen, Geschäftsnetzwerke; mobile und ortsbezogene Informationsdienste; Webdienste, ...)

## Software-Produktqualität (siehe auch ISO 9126/DIN 66272)

- **Zuverlässigkeit**  
Software darf im Fall des Versagens keine physischen oder ökonomischen Schäden verursachen.
- **Benutzbarkeit**
  - Software muss sich nach den Bedürfnissen der Benutzer richten.
  - Die Benutzerschnittstelle muss ergonomisch und selbsterklärend sein.
  - Dokumentation muss in allen Detaillierungsgraden ausreichend zur Verfügung stehen.
- **Wartbarkeit**
  - Software muss anpassbar an neue Anforderungen sein.
  - Software sollte möglichst plattform-unabhängig sein.
- **Effizienz**
  - Software muss ökonomischen Gebrauch von Ressourcen des unterliegenden Systems machen.

## Außensicht der Anwender: Software als Produkt

- **Produkt**  
Ein in sich abgeschlossenes, i. a. **für einen Auftraggeber** bestimmtes Ergebnis eines erfolgreich durchgeführten Projekts oder Herstellungsprozesses [Balzert 2000]
- **Softwareprodukt**
  - **Individualsoftware** entwickelt für einen Auftraggeber (z.B. Kundenverwaltung, Hubschraubersteuerung)
  - **Standardsoftware** entwickelt für Anwenderklasse, z.B. Büro (MS-Office), Logistik (SAP R/3),
  - **Typische Systemarten**
    - Informationssysteme (Sachbearbeitung)
    - Eingebettete Systeme (Steuerungen)
    - Produktionsplanung und -steuerung
    - Telekommunikation (Vermittlungssysteme)
    - E-Business (Web-Systeme)

## Software-Produktqualität (ISO 9126/DIN 66272)

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>▪ <b>Funktionalität</b> <ul style="list-style-type: none"> <li>▪ Angemessenheit</li> <li>▪ Sicherheit</li> <li>▪ Genauigkeit der Berechnung</li> <li>▪ Interoperabilität</li> <li>▪ Konformanz zu Standards</li> </ul> </li> <li>▪ <b>Zuverlässigkeit</b> <ul style="list-style-type: none"> <li>▪ Reife</li> <li>▪ Fehlertoleranz</li> <li>▪ Wiederherstellbarkeit</li> </ul> </li> <li>▪ <b>Benutzbarkeit</b> <ul style="list-style-type: none"> <li>▪ Verständlichkeit</li> <li>▪ Erlernbarkeit</li> <li>▪ Bedienbarkeit</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>▪ <b>Effizienz</b> <ul style="list-style-type: none"> <li>▪ Zeitverhalten</li> <li>▪ Verbrauchsverhalten</li> </ul> </li> <li>▪ <b>Änderbarkeit</b> <ul style="list-style-type: none"> <li>▪ Analysierbarkeit</li> <li>▪ Modifizierbarkeit</li> </ul> </li> <li>▪ <b>Stabilität</b> <ul style="list-style-type: none"> <li>▪ Prüfbarkeit</li> </ul> </li> <li>▪ <b>Übertragbarkeit</b> <ul style="list-style-type: none"> <li>▪ Anpassbarkeit</li> <li>▪ Installierbarkeit</li> <li>▪ Konformanz zu Standards</li> <li>▪ Austauschbarkeit</li> </ul> </li> </ul> |
|---|---|

Block R (Rahmen): Einführung

## Was ist Software Engineering?

---

- Eine naive Sicht:
 

Problem Spezifikation
→
*Kodierung*
→
Programm
- Aber ...
  - Wo kommt die Spezifikation her?
  - Woher weiß man, dass die Spezifikation die Nutzerbedürfnisse erfüllt?
  - Wie wurde die Struktur des Programms bestimmt?
  - Woher weiß man, dass das Programm die Spezifikation erfüllt?
  - Woher weiß man, dass das Programm immer zuverlässig arbeiten wird?
  - Was macht man, wenn der Anwender Änderungen benötigt?
  - Wie strukturiert man die Programmentwicklung in Unteraufgaben, wenn ein Team von Entwicklern zur Verfügung steht ?

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Software Engineering (III)

---

*“multi-person construction of multi-version software”*  
*David L.Parnas*

- Teamarbeit
  - Aspekt der Größenordnung (“Gut programmieren” ist nicht genug)
  - Kommunikationsaspekt
- Erfolgreiche Softwaresysteme müssen sich weiterentwickeln oder sie werden verschwinden
  - Änderung ist die Norm, nicht die Ausnahme

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Software Engineering (I)

---

Einige Definitionen und Aspekte

- *The establishment and use of sound engineering principles in order to obtain economically software that is reliable and runs on real machines.*  
*F.L. Bauer, NATO-Konferenz Software-Engineering 1968*
- *The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*  
*IEEE Std. 610.12 (1990)*

Also:

- Bereitstellung und systematische Anwendung von Methoden, Verfahren und Werkzeugen zur Entwicklung, Betrieb und Wartung von Software.

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Software Engineering (IV)

---

*“software engineering is different from other engineering disciplines”*  
*Ian Sommerville*

- Nicht durch physikalische Gesetze eingeschränkt
  - Die Grenze ist der menschliche Geist
- Aber beschränkt durch politische Kräfte
  - Interessen der Beteiligten abwägen und ausgleichen

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Software Engineering (I)

---

*“state of the art of developing quality software on time and within budget”*

- Trade-off zwischen Perfektion und physischen Grenzen  
SE muss sich mit Problemen der realen Welt auseinandersetzen
- Stand der Kunst!
  - Die Gemeinschaft entscheidet über „Beste Praxis“
  - Lebenslange Weiterbildung

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Professionelle Software-Entwicklung

---

- Planung & Aufwandsschätzung
- Teamarbeit & Verantwortung
- Qualität
- Benutzerfreundlichkeit
- Wartungsfreundlichkeit & Anpaßbarkeit
- Wirtschaftlichkeit
- Sicherheit
- ...

Ein großes “Software-Gebäude” ist nicht nur ein etwas größeres Programm. Es sind völlig andere Technologien zur Herstellung und Wartung nötig.

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Methoden des Software Engineering: WS 2003/04

**Dozenten:**  
Dr. Nora Koch, Dr. Harald Störrie, Prof. Dr. Martin Wirsing

**Übungsleitung:**  
Dipl. Inf. Florian Hacklinger

**Vorlesung** strukturiert in einzelne Moduln

**Folien** möglichst am Vorabend der Vorlesung im Netz  
<http://www.pst.informatik.uni-muenchen.de/lehre/WS0405/mse/>

**Feedback**  
Stellen Sie **viele Fragen** während der Vorlesung!  
Ohne **Rückmeldung** können wir nichts verändern!  
Durchführung: Am Besten **konstruktiv & offline**.

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Software-Engineering an der LMU

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Übungen zur Vorlesung

**Voraussetzungen:**  
Gute Programmierkenntnisse (möglichst Java)

**Übungsbetrieb:**  
Wöchentliche Übungen, Mi 16-18 Uhr, Oettingenstr. 1.14, Beginn 2. Woche  
3 Hausaufgabenblätter  
Klausur

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Inhalt der Vorlesung

- Block R:** Einführung und Übersicht (MW)
- Block A:** Anforderungsentwicklung (MW)
- Block B:** Software-Architektur (MW/HS)
- Block C:** Formale Methoden (MW)
- Block D:** Validierung & Verifikation (MW)
- Block E:** Software-Prozess & Projektmanagement (MW/HS)
- Block F:** Web-Engineering (NK)
- Block R:** Ausblick (MW)

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Literatur

- Ian Sommerville: Software Engineering, 6th edition, Addison-Wesley 2000.
- Helmut Balzert: Lehrbuch der Software-Technik (2 Bände), Spektrum Akademischer Verlag 2000 und 1998.
- Weitere Literatur bei den einzelnen Moduln

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Anforderungsanalyse

- Anforderungserwerb
- Konzeptuelle Modellierung
- Anforderungsvalidierung

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Software-Architektur

- Architekturdentwurf
- Systemarchitektur
- Middleware
- Entwurfsmuster & Komponenten

**Beispiel:**  
Architektur einer Java-Entwicklungs-plattform (informell)

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Prozess und Projektmanagement

- Vorgehensmodelle
- Prozessverbesserung
- Projektmanagement
- Empirisches Software-Engineering

**Beispiel:** Wasserfallmodell - vereinfachte Darstellung

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Formale Methoden

- Funktionale Spezifikation
- Eigenschaften dynamischer Systeme
- Zustandsautomaten
- Petrinetze

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Web-Engineering

- Modellierung von Web Anwendungen
- Web-Architektur, -Implementierung und Testen
- Web-Prozessmodelle und Projektmanagement

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Validierung und Test

- Testplanung
- Blackbox- und Whiteboxtest
- Modultest mit JUnit
- Verifikation durch Modelchecking

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München

Block R (Rahmen): Einführung

## Zusammenfassung

- Software-Produktqualität umfasst die Aspekte  
Funktionalität, Zuverlässigkeit, Benutzbarkeit,  
Effizienz, Änderbarkeit, Übertragbarkeit
- Software Engineering bedeutet  
die Bereitstellung und systematische Verwendung von Methoden, Verfahren  
und Werkzeugen zur Entwicklung, Betrieb und Wartung von Software

Methoden des Software Engineering WS04/05, N. Koch, H. Störrie, M. Wirsing, LMU München