Software Engineering

Einleitung, Motivation, Definition

Prof. Dr. Peter Jüttner

Hochschule Deggendorf

Vorbemerkungen

- Große Teile der Unterlagen zu dieser Vorlesung entstanden im Rahmen einer Lehrveranstaltung an der Hochschule Regensburg

- Co-Autor war damals Hr. Dr. Christian Flug, Continental AG, dem ich an dieser Stelle nochmals herzlich für die hervorragende Zusammenarbeit danken möchte!

Frage

Software Engineering?

Ziele der Vorlesung

- **Die wichtigen Begriffe des industriellen Software Engineering einführen**
- **Die wichtigen Methoden des industriellen Software Engineering kennen lernen**
- **Methoden üben**
- **Teamarbeit üben**

Inhalte

- **Software: Definitionen, Anforderungen und Eigenschaften**
- **Motivation SW Engineering**
  - **SW Krise**
  - **Bekannte SW Probleme**
- **Definitionen SW Engineering**
- **Elemente des SW Engineering und deren Ziele, Inhalte**
  - **Engineering Phasen**
  - **Unterstützende Prozesse**
- **Vorgehensmodelle / Prozeßmodelle**
- **SW Engineering Methoden**

Inhalt

1. Einleitung

2. Motivation und Definition

2.1. Warum SW Engineering

2.2. Definitionen SW Engineering

Inhalt

3. Elemente des SW Engineering und deren Ziel

3.1. Requirements

3.2. Architektur & Feindesign

3.3. Codierung

3.4. Test

3.5. Projekt-Management

3.6. Rollen

3.7. Qualitätssicherung

3.8. Konfigurations-Management

3.9. Versionsmanagement

3.10. Änderungsmanagement

Inhalt

3.11. Notationen

3.12. Dokumentation, Templates und Beispiele

3.13. Methoden

3.14. Tools

4. Vorgehensmodelle / Prozeßmodelle (Beschreibung, Vorteile, Nachteile, Bewertung)

4.1. Wasserfall

4.2. V-Modell

4.3. Iterative / Agile Methoden

4.3.1. Extreme Programming

Inhalte

4.3.2 Scrum

4.3.3. RUP

4.4. Prozessbewertung

4.4.1. CMMI

4.4.2. SPICE

Inhalte

5. Methoden

5.1. Requirements Engineering

5.2 Architektur & Feindesign

5.3 Implementierung

5.4 Test

5.5 Qualitätssicherung

5.6 Projektmanagement

5.7 Konfigurationsmanagement

6. Der Faktor Mensch in der SW Entwicklung

Zeitplan

- **Mittwoch 8:00 – 11:15 Uhr Vorlesung**
- **Mittwoch 12:00 – 13.30 Uhr Praktikum**

- **Sprechstunde Montag 11.15 Uhr, Raum E225**

Übungen & Praktikum

- Kleinere Trockenübungen während der Vorlesung
- Praktikum mit konkreter Teamaufgabe
- Zu bearbeiten ist eines der folgenden Themen:
  - Fußball
  - Software für ein Karosseriesteuergerät (Bordcomputer)
  - Entwicklung eines Projektmanagementtools
  - eigenes Thema nach Rücksprache mit Dozent
- Für das Projekt ist ein Projektantrag (Thema, Team) zu erstellen

Praktikum

- Ziel des Praktikums ist es, ein kleines Projekt in einem Projektteam zu bearbeiten.
- Die Teamgröße 3 - 5 Personen
- Im Projekt sollen Rollen definiert und (formell / informell) eingenommen werden:
  - Projektleiter
  - System-Ingenieur /Requirements Ingenieur
  - Architekt/Implementierer
  - Tester
  - Qualitäts-Ingenieur
  - …

Praktikum

- Eine Person kann mehrere Rollen einnehmen. Zusätzlich sollen die jeweils anderen Projektmitglieder dem jeweils Verantwortlichen zuarbeiten (z.B. sollen alle implementieren, aber eine Person ist für das Gesamtresultat bzw. das zugehörige Arbeitsergebnis verantwortlich)

Praktikum

- Bewertung der Praktikums
  - Jedes Projektmitglied übernimmt die Verantwortung für **ein individuelles** Arbeitsergebnis (Dokument) der Prakitkumsaufgabe (z.B. Plaung, Architektur, Testspezifikation)
    Abgabe -> s. Dokument auf Vorlesungslaufwerk
  - Präsenzveranstaltung (min. 10 Praktikumstage)

Klausur

- 90 Min.
- Keine Unterlagen erlaubt!
- Klausuraufgaben gemäß Vorlesungsübungen und Projektarbeit
- Vorlesungs-, Übungsinhalt und Transfer
- Probeklausur
- Teilnahme an der Klausur **nur nach** erfolgreich absolviertem Praktikum

Zum Schluss dieses Abschnitts …

Noch Fragen ??

Inhalt

2. Motivation und Definition

2.1. Warum SW Engineering

- Software
- Anforderungen an SW
- Software Krise,
- Probleme, Unglücke, Katastrophen, die auf Grund von SW-Fehlern verursacht wurden

2.2. Definitionen SW Engineering

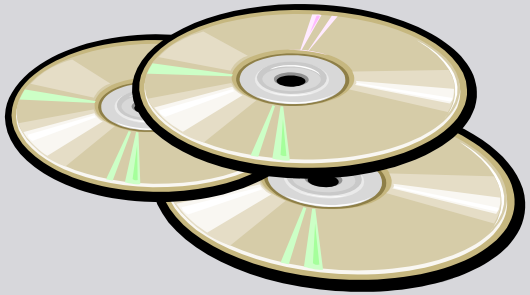
- Zitate, Ziele, Merkmale

Software

Software:

- Computer programs, procedures, rules, and possibly associated documentation and data pertaining to the operation of a computer system. (IEEE Standard Glossary of Software Engineering)

- Computer Programme, Abläufe, Regeln, und möglicherweise zugehörige Dokumentation und Daten, die zum Ablauf eines Computersystems gehören.



Software

Klassifikationen von Software

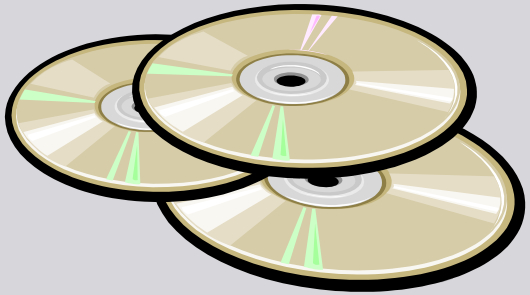
- ?

Eigenschaften von Software

- ?

Anforderungen an Software

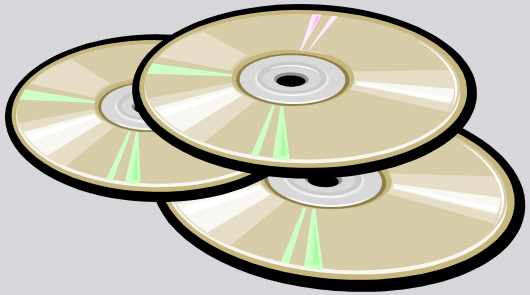
- ?



Software

Klassifikationen von Software

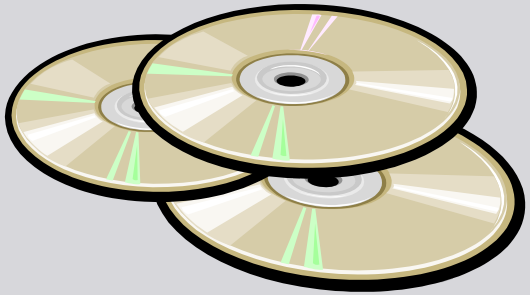
- Software als eigenständiges Produkt (z.B. für PC, Handy)
- Software als Bestandteil eines Produkts (z.B. eingebettete Software)
- Generisches Produkt oder Einzelanfertigung (vereinbartes Produkt)
- Echtzeitsoftware (z.B. für eingebettete Systeme)
- Verteilte Software
- Monolithische Software
- Betriebssystemsoftware
- Kommunikationssoftware
- Applikationssoftware
- Grafische Software
- …



Software

Eigenschaften von Software

- Software ist nicht fassbar
- Software nutzt sich nicht ab
- Software veraltet (auch ohne Abnutzung)
- Software wird in identischen Exemplaren ausgeliefert
- Software hat (theoretisch) keine Grenzen
- Software ist meist zu teuer
- Software ist oft / meist / immer fehlerhaft



Software

Anforderungen an Software

- Korrekt (fehlerfrei oder wenigstens fehlerarm ?)
- Portierbar
  - verwendbar in verschiedenen Umgebungen
- Wartbar
  - Fehler sind schnell korrigierbar
- Erweiterbar
  - Neue Funktionen sind leicht integrierbar

Software

Anforderungen an Software

- Robust (z.B. gegen Störungen)
  - Externe Störungen beeinflussen Funktion nicht
- Verfügbar
  - Bei Bedarf
  - Ggf. änderbar im laufenden Betrieb
- Funktional Sicher
  - Schutz vor körperlichen und/oder finanziellen Schäden
- Datensicher
  - Schutz vor Datenverlust bzw. Verfälschung

Software

Anforderungen an Software

- Zugriffssicher
  - Schutz vor Benutzung durch Nicht-Autorisierte
- Klein
  - geringer Speicherplatzverbrauch
- Schnell
  - kurze Reaktionszeit
- Testbar
  - Ausreichende Testabdeckung mit akzeptablem Aufwand erreichbar

Software

Anforderungen an Software

- Nachvollziehbar
  - Entwicklungsschritte dokumentiert und erkennbar
- Zertifizierbar
  - Genügt bestimmten Standards (z.B. funktionale Sicherheit, Integration in bestimmte Umgebungen, Compiler)
  - kann auch den Entwicklungsprozess betreffen
- Dokumentiert
  - Nicht nur der Code ist vorhanden
- Zuverlässig
  - funktioniert immer (gleich in gleicher Umgebung)

Software

Anforderungen an Software

- Benutzbar
  - Software ist an die Bedürfnissen der Benutzer angepasst
  - Die Benutzerschnittstelle muss ergonomisch und selbsterklärend sein.
  - Das Benutzermanual muss in der notwendigen Detaillierung zur Verfügung stehen
- Effizient
  - wirtschaftliche Nutzung der zur Verfügung stehenden Ressourcen der Umgebung.

Software

Anforderungen an Software

- Kompatibel
  - d.h. arbeitet mit anderer SW zusammen
- Kostengünstig
- Schnell verfügbar
- Messbar
  - während der Entwicklung
  - als Produkt
- …

Software

Achtung:

Diese Anforderungen widersprechen sich teilweise, d.h. es sind nicht immer alle gleichzeitig erfüllbar!

Fehlerfreiheit ist nie erfüllbar… ?

Software

Achtung:

Diese Anforderungen bedingen sich teilweise, d.h. um manche zu erfüllen, müssen andere erfüllt sein!

Übung

Beispiele für widersprüchliche Anforderungen

Identifizieren Sie Anforderungen an Software, die sich widersprechen und Anforderungen, die sich bedingen.

Begründen Sie Ihr Ergebnis!

10 min, arbeiten Sie ggf. zusammen mit einem Partner

Software

D.h.:

Es müssen diejenigen Anforderungen ausgewählt werden, die für die zu entwickelnde Software maßgeblich sind.

Sind widersprüchliche Eigenschaften zu erfüllen, so ist ein geeigneter Kompromiss zu finden.

Motivation

- Juli 1962, Verlust der ersten amerikanischen Venussonde Mariner 1
  Entscheidender Fehler: Punkt statt Komma im Fortran Sourcecode!

- Januar 1990: 70 Millionen von 138 Millionen Ferngesprächen innerhalb USA konnten 9 Stunden lang nicht vermittelt werden.
  ➔ Schaden ca. 75 Mio $, bei AT&T (ohne Folgeschäden), mindestens so viel bei AT&T Kunden,
  Ursache: Ein break-Befehl von C wurde falsch eingesetzt (Der Fehler wurde eine Woche später gefunden, existierte seit einer Programm-Optimierung 4 Wochen vorher)

Motivation

- F-16 Jagdflugzeug, ca. 1978: Erstes Jagdflugzeug der USA mit Computer-Steuerung. Beim Überfliegen des Äquators stellt sich das Flugzeug auf den Kopf wegen eines Vorzeichenfehler im Algorithmus des Programms bei der Berücksichtigung der geographischen Breite (Fehler wurde rechtzeitig beim Test gefunden)

- Nimrod-Luftüberwachungssystem (UK):
  - Probleme: unklare Anforderungen, …
  - Entwicklungszeit: ca. Mitte 70er bis Mitte 80er Jahre
  - Entwicklungskosten: mehrere hundert Million Pfund
  - Ergebnis: durch AWACS ersetzt

Motivation

- August 1988: 1. Marssonde der UdSSR: Beim Senden eines Kommandos kurz nach dem Start wurde ein Buchstabe vergessen. Das verkürzte Kommando wurde jedoch als ein anderes (nur beim Bodentest vorgesehenes) Kommando interpretiert. Folge war eine Rotation der Raumsonde, so dass der Kontakt mit ihr abbrach. (Auch die Schwester-Sonde Phobos 2 ging später (beim Marsmond Phobos) wegen eines Software-Fehlers verloren)

- 1994: Eröffnung des Denver International Airport um 9 Monate verzögert wegen Softwareproblemen im Gepäcktransport-System
  ➔ Schaden ?

Motivation

- 1996 Ariane 5 - Selbstzerstörung auf Grund eines SW Fehlers (Wiederverwendung einer SW aus Ariane 4 in "unpassender" Umgebung, Geschwindigkeit 5mal höher -> Überlauf)
  ➔ Schaden ca. 500 Mio $

- März 1999: Fehlstart einer Titan/Centaur-Rakete wegen falscher Software-Version

- September 1999: Verlust der Sonde "Mars Climate Orbiter" wegen falscher Einheitenumrechnung (metrisch / Fuss)

- US-Unternehmen verloren im Jahr 2000 insgesamt 100 Milliarden US-$ wegen defekter Software.

Motivation

- Verzögerte Auslieferung von ICE-Zügen an die Deutsche Bahn wegen u.a. SW-Problemen

Motivation

Software Krise:

Mitte der 60er Jahre verbreitet sich der Begriff der ***Software Krise*** in Industrie und Wissenschaft

Zahlreiche Untersuchungen folgen über Projekte, die

- teurer als geplant (um Faktoren)
- später fertig als geplant
- nie fertig

wurden, wurden durchgeführt

Motivation

z.B. US Regierungsstatistik von 1979 (9 Projekte):

- für 3.2 Mrd. US $ Software wurde nie ausgeliefert
- für 2.0 Mrd. US $ Software wurde ausgeliefert, aber nie benutzt
- für 1.3 Mrd. US $ Software wurde nur kurzzeitig benutzt
- für 0.2 Mrd. US $ Software wurde mit Änderungen nach Auslieferung enutzt
- für 0.1 Mrd. US $ Software wurde ohne Änderungen nach Auslieferung benutzt

Motivation

„Als es noch keine Rechner gab, war auch das Programmieren noch kein Problem, als es dann ein paar leistungsschwache Rechner gab, war das Programmieren ein kleines Problem und nun, wo wir gigantische Rechner haben, ist auch das Programmieren zu einem gigantischen Problem geworden. In diesem Sinne hat die elektronische Industrie kein einziges Problem gelöst, sondern nur neue geschaffen. Sie hat das Problem geschaffen, ihre Produkte zu nutzen.“
Edsger W. Dijkstra (1972)

Zum Schluss dieses Abschnitts …

Noch Fragen ??

Inhalt

2. Motivation und Definition

2.1. Warum SW Engineering

- Software
- Anforderungen an SW
- Software Krise,
- Probleme, Unglücke, Katastrophen, die auf Grund von SW-Fehlern verursacht wurden

2.2. Definitionen SW Engineering

- Zitate, Ziele, Merkmale

Software Engineering

Der (bzw. ein) Weg aus der Krise:

Software Engineering

Software Engineering

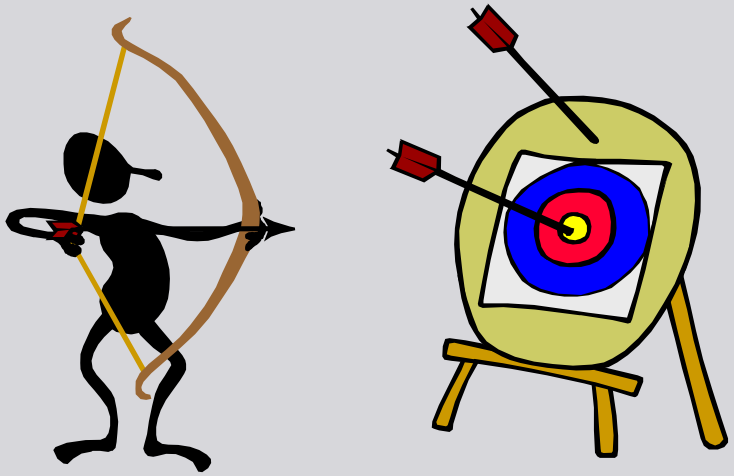
Software Engineering = Software Technik

Definitionen Software Engineering

Ziel des Software Engineering:

Sicherstellen, dass die Anforderungen an die Software im Rahmen der Entwicklung auf systematische Weise erfüllt werden!

Dazu werden im Rahmen des Software Engineering geeignete Mittel (z.B. Prozesse, Methoden) angewendet.



Definitionen Software Engineering

- Software Engineering: 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)

- Software Engineering: The practical application of scientific knowledge in the design and construction of computer programs and the associated documentation required to develop, operate, and maintain them (Berry Boehm 76)

- Software-Technik ist die zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden, Konzepten, Notationen und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Software-Systemen. (Balzert)

Definitionen Software Engineering

Eine *technische Disziplin*, die sich mit *allen Aspekten der Softwareherstellung* befasst, von den frühen Phasen der Systemspezifikation bis hin zur Wartung des Systems, nachdem sein Betrieb aufgenommen wurde.

Technische Disziplin

- Suche nach Lösungen (auch, wenn keine theoretischen Grundlagen existieren) innerhalb organisatorischer und finanzieller Beschränkungen
- Alle Aspekte der Softwareherstellung
- Nicht nur technische Aspekte, sondern auch Projektverwaltung, Entwicklung neuer Theorien, Methoden, Werkzeuge etc.

(Ian Sommerville)

Definitionen Software Engineering

Die *IEEE Computer Society* definiert **Software Engineering** als

The application of a

- systematic,
- disciplined,
- quantifiable

approach to the

- development,
- operation, and
- maintenance

of software;

that is, the application of engineering to software.

Definitionen Software Engineering

Software Engineering bedeutet:

- Ingenieurmäßiges Vorgehen
- Wissenschaftliche Erkenntnisse als Basis
- Empirische Erkenntnisse als Basis
- Wirtschaftliches Vorgehen
- Lauffähige Software als Ergebnis
- Zuverlässige Software
- Unterstützung von Entwicklung, Betrieb und Wartung (Lebenszyklus)

Definitionen Software Engineering

Software Engineering bedeutet:

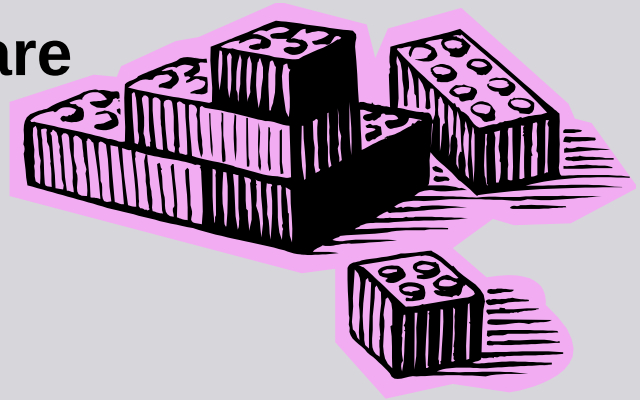
- Die Entwicklung wird vorhersagbar(er) bezüglich Ergebnis, Kosten, Zeit, Qualität, …
- Die Entwicklung wird messbar(er) bezüglich Ergebnis, Kosten, Zeit, Qualität, …
- Die Entwicklung wird wiederholbar (mit dem gleichen oder ähnlichen Ergebnis)
- Unabhängigkeit von Personen
- Vermittelbarkeit (in Forschung und Lehre)
- ...

Definitionen Software Engineering

Achtung:

Software Engineering stellt nur einen Methoden-Baukasten bereit, mit dem Software entwickelt werden kann.

Für ein konkretes Projekt sind die geeigneten Elemente aus dem Software Engineering Baukasten auszuwählen



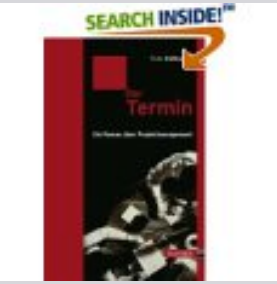
d.h. der Baukasten ist richtig anzuwenden

Definitionen Software Engineering

Mit dem Software Engineering entsteht auch der Software Engineer (= Software Ingenieur)

Literatur Software Engineering

Ian Sommerville, Software Engineering, Pearson Studium, ISBN: 0321210260



Tom de Marco, Der Termin, Hanser Fachbuchverlag, Leipzig 1998, ISBN 3446194320

Helmut Balzert, Lehrbuch der Software-Technik, Band 1. 2. Auflage. Elsevier-Verlag, 2001

Motivation

Zum Schluss dieses Abschnitts …

Noch Fragen ??