Qualitätsmanagement von Software und Systemen

Einführung und Überblick

Grundlagen des Software Engineering Einführung und Überblick

- Eckdaten der Softwarebranche in Deutschland
- Was ist Software?
- Der Wandel im Automobilbereich
- Mariner 1 und Ariane 5
- Was ist Softwaretechnik?
- Ziele der Lehrveranstaltung

Beispiele: Verbreitung von Softwaresystemen

- Haushalts- und Konsumelektronik
  - „Einfachste“ Geräte wie Kaffeemaschinen, Waschmaschinen und Kühlschränke beinhalten Softwaresysteme
  - Moderne Geräte wie Handys, DVD-Player und Digitalkameras bestehen zum größten Teil aus Software
- Automobilindustrie
  - Betriebliche Abläufe, Verwaltung und Produktion wäre ohne Softwaresysteme nicht mehr möglich
  - In einem Fahrzeug sind heute ca. 100 Mikrocontroller integriert
  - Mehr als die Hälfte aller Fahrzeugpannen lassen sich auf Softwareprobleme zurückführen
- Informationssysteme
  - Anwendungsbranchen: Finanzwesen, Medizinwesen, Verwaltung, …
  - Informationssysteme haben inzwischen einen Durchdringungsgrad in der Geschäftsprozessunterstützung von 60% bis 90%
  - Die Abwicklung eines Geschäftsprozesses erfordert unter Umständen das Zusammenspiel von mehr als 15 Großanwendungen

Software-Katastrophe: Patriot-Rakete (1)

Ein fataler Software-Fehler im Golfkrieg II:

"During the Gulf war, a computer failure was responsible for the failure of a patriot missile to stop a scud missile that hit an American military barracks in Dharan ... 28 dead ..."

[Quelle: ACM SIGSOFT Software Engineering Notes, vol. 16, no. 3 (1991), S.19f]

Ursache:

- der Steuercomputer lief 4 Tage ununterbrochen (statt der vorgeschriebenen maximal 14 Stunden)
- dadurch lief das interne Timer-Register über 24 Bit hinaus und es entstanden Rundungsfehler bei der Bahnberechnung
- wäre das Timer-Intervall 1/8 statt 1/10 Sekunde gewesen hätte es keine Rundungsfehler gegeben
- das Intervall wurde entgegen der ursprünglichen Programmierung nachträglich von einem Manager auf 1/10 Sek. geändert

Software-Katastrophe: Patriot-Rakete (2)

Schlussfolgerungen aus dem „Patriot-Missile“ -Beispiel:

- Software soweit wie möglich gegen Fehlbedienungen absichern (z.B. Warnungen nach 14 Stunden Laufzeit)
- Software soweit wie möglich gegen typische Programmierfehler absichern (Zählerüberläufe etc. durch geeignete Plausibilitätsprüfungen und „exception handling“ abfangen)
- Wichtige Entwurfsentscheidungen sind für spätere Wartung zu dokumentieren („1/8 Sek. Timer-Intervall wurde gewählt, weil…“)
- Für Software-Entwicklung feste Vorgehensweisen und Zuständigkeiten festlegen (um ad-hoc Änderungen durch unqualifizierte Personen zu verhindern)

[Quelle: Mark Minas, Vom Bild zum Programm, S.12f]

Software-Katastrophe: Kein Einzelfall (1)

- 1981: US Air Force Command & Control Software überschreitet Kostenvoranschlag fast um den Faktor 10: ***3,2 Mio. US-$***.
- 1987-1993: Integration der kalifornischen Systeme zur Führerschein- und KFZ-Registrierung abgebrochen: ***44 Mio. US-$***.
- 1992: Integration des Reservierungssystems SABRE mit anderen Reservierungssystemen abgebrochen: ***165 Mio. US-$***.
- 1997: Entwicklung des Informationssystems SACSS für den Staat Kalifornien abgebrochen: ***300 Mio. US-$***.
- 1994: Eröffnung des Denver International Airport um 16 Monate verzögert wegen Softwareproblemen im Gepäcktransport-System: ***655 Mio. US-$***.
- 2005: Das deutsche Maut Erfassungssystem "Toll Collect" konnte nur mit erheblicher Verzögerung (Vertragsabschluss: September ‘02, geplanter Starttermin: 31. August 2003), am 1. Januar 2005 in technisch reduzierter Form in Betrieb genommen werden: ***~6,5 Mrd. €***.

Software-Katastrophe: Kein Einzelfall (2)

- 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 abgehoben 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.

Zunehmende QS-Anforderungen

- Für 50% des Ausfälle im industriellen Sektor sind Software-Fehler verantwortlich
- Schwierigkeiten mit Zuverlässigkeit durch hohe Komplexität
  - p_k: Wahrscheinlichkeit, dass eine Komponente nicht fehlerhaft ist
  - p_s: Wahrscheinlichkeit, dass das System nicht fehlerhaft ist

Anzahl Komponenten	p_k	p_s
10	0,9	0,35
10	0,99	0,9
100	0,9	0,000027
100	0,99	0,37

- Fehler in 1.000 LOC
  - 1977: 7 - 20
  - 1994: 0,05 – 0,2
- Durchschnittliche Programmgröße (in 1.000 LOC)
  - 1977: 10
  - 1994: 800

IT-Katastrophen – nur Einzelfälle?

- 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.

Erfolgsstatistik von IT-Projekten

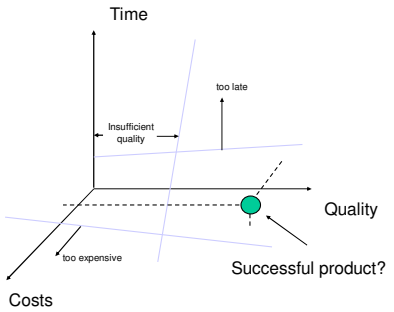
	Succeeded	Failed	Challenged
1994	16%	31%	53%
1996	27%	40%	33%
1998	26%	28%	46%
2000	28%	23%	49%

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

Was ist Software Engineering? Definition

- Software Engineering ist die zielorientierte Bereitstellung und Verwendung von **systematischen, ingenieurmäßigen und quantifizierbaren Vorgehensweisen** für **Entwicklung, Betrieb, Wartung und Stilllegung von Softwaresystemen**
- Zielorientiert bedeutet dabei die Berücksichtigung von
  - Zeit
  - Kosten
  - **Qualität**

Motivation
Der Zielkonflikt für erfolgreiche Produkte



Rechtliche Verantwortung

Ausschnitt aus Safety - Handbuch der BW
SIL Safety Integrity Level

HOCH NIEDRIG

Attributes	SIL 4	SIL 3	SIL 2	SIL 1	Appl. HW SW
Requirements and Design Specification	Formal (Mathe-matical)	Semiformal	Informal (e.g. Natural Language)	Informal (e.g. Natural Language)	H/S
Configuration Management	Full (Automated for development and production	Full (Automated for development and production	Yes	Manual	H/S
Structured Design Method; e.g. data	Yes	Yes	Preferred	Optional	H/S

Verpflichtend durchzuführen-de Aktivitäten

Werden diese nicht durchge-führt ist der Ingenieur haft-bar!

Gilt auch wenn nicht nach State-of-the-art entwickelt wird.

Die Bedeutung von Sicherheitsnachweisen und Zuverlässigkeitsanalysen

- Sicherheitsnachweise durch gesetzliche Regelungen oder Zulassungsstellen gefordert, z. B.:
  - Schienenverkehr: EBA (Deutschland)
  - Medizintechnik: FDA (USA)
- Zuverlässigkeitsziele zunehmend von Kunden gefordert (z. B. Automobilindustrie)
- Verfügbarkeitsanforderungen als Vertragsbestandteil mit Konventionalstrafen belegt (z. B. öffentliche Vermittlungstechnik, Schienenverkehrssysteme)
- Produkthaftungsgesetz sieht eine umfassende Haftung von Herstellern vor (und definiert den Begriff der „Herstellers“ sehr weit)

Qualitätsmanagement

- Definition der Prozesse im Hinblick auf die Erreichung von Qualitätszielen
- Festlegung geeigneter Techniken zur „Konstruktion von Qualität“
- Beschreibung geeigneter Kontrollverfahren zur Qualitätsanalyse- und Qualitätsmessung
- Schaffung von Auswertungstechniken für die Analysedaten
- Einbindung aller Mitarbeiter und Manager entsprechend ihrer Zuständigkeiten
- Etablierung eines Verfahrens für die ständige Überwachung und Verbesserung der genannten Aspekte