# Software Engineering

SoSe18 Bulenda

Grundlagen

1.0 . Orga

* 1. Was ist Software und wo?
  2. Zielorientierung
  3. Warum braucht man SE?

## 1.0.Orga

* Klausur: 90 min
* Einschreibeschlüssel 2357

## 1.1.Einleitung

Arten von Software:

* Produkt-oder Individualentwicklung
* Systemsoftware (Betriebssystem, Compiler, Editor,…) oder Anwendungssoftware
* Produktintegriert oder für reine Computersysteme
* Datenintensiv oder berechnungsintensiv
* Monolithisch oder verteilt
* Standaloneoder mit anderen Anwendungen integriert

Eigenschaften von Software:

* Software ist immateriell.
* Software unterliegt keinem Verschleiß.
* Software veraltet.
* Es gibt keine Software Ersatzteile: Defekte sind immer Konstruktionsfehler.
* Software ist schwer zu vermessen.
* Software gilt als leicht änderbar.
* Software unterliegt einem ständigen Anpassungsdruck.

Software Engineering teilt sich in drei Bereiche:

* Software-Management

Planung, Organisation, Leitung und Kontrolle einer Software-Entwicklung.

* Software-Entwicklung

Aus einem geplanten Software-Produkt ein fertiges Software- Produkt entwickeln, das die geforderten Eigenschaften besitzt.

* Software-Qualitätssicherung

Sicherstellung der geforderten Produkt- und Prozessqualität einer Softwareentwicklung durch geeignete konstruktive und analytische Maßnahmen.

Merkmale schlechter Software:

* **Steifheit** (Rigidity): selbst kleine Änderungen sind schwierig, aufwändig und riskant
* Symptom: State of the art fkt. werden nicht eingebaut., nicht kritische Fehler bleiben beunrücksichtigt
* **Zerbrechlichkeit**(Fragility): kleine Fehler lösen Fehler/anomalien in entfernten Softwareteilen aus
* Symptom: Großteil(bis zu 90%) der Entwicklungsarbeit geht in Bugfixing
* **Unbeweglichkeit** (Immobility): viele Querverbindungen/Abhängigkeiten, SW kann weder als Ganzes noch in Teilen wiederverwendet werden
* Symptom:Parallele Produktlinien mit ähnlicher Funktionalität und unabh. Code Basis
* **Zähigkeit** (Viscosity): Änderungen gegen den Entwurfsgedanken sind leichter als Änderungen im Sinne des Entwurfsgedanken.
* Symptom: Nicht genutzer Code, irrelevante Doku, redunanter Code

1.2 Zielorientierung

Die Erfüllung wirtschaftlicher Ziele ist das Ziel :D

Qualität

Software-Produkt-Qualität ist die Gesamtheit von Eigenschaften eines Software Produkts, die sich auf die Eignung zur Erfüllung gegebener Erfordernisse beziehen.

Die Erfordernisse des Anwenders sind entscheidend.

Kosten/Aufwand

Es gibt 5 Hauptkostenfaktoren, die zu berücksichtigen sind.

Zum ersten gibt es da natürlich die **Entwicklungskosten** (PC, Personal…), zudem

Kommen Kosten für die **Wartung**, **Fehlerbeseitigung** (Anpassungen, Erweiterungen)

Sowie Kosten für die **organisatorische** **Implementierung**(Schulungen z.B.) sowie zu

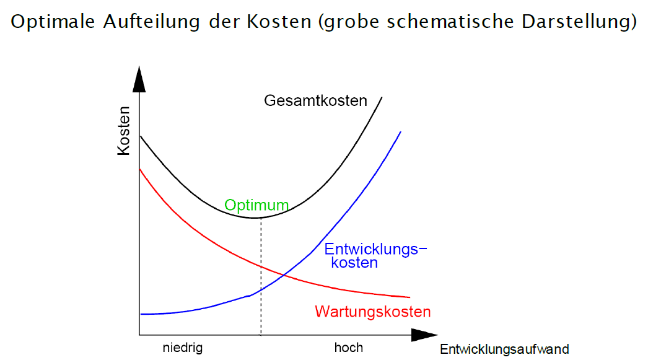
Guter Letzt **Vertriebskosten** (Werbung, Personal Selling…)

Prozedurale Aufteilung der Kosten:

Die Gesamtkosten bestehen aus ca. **33% Entwicklungskosten** **und 67% Wartungskosten**.

Die Entwicklungskosten setzen sich aus ca **40% Analyse/Entwurf, 40% Testen** **und 20% Codierung** zusammen.

**Daraus folgt:** wenig Wartung bedeutet enorme Kosteneinsparung.



Zeit/Termine

Entwicklung soll natürlich möglichst schnell gehen, damit man das Produkt schnell aufn Markt werfen kann. Natürlich gibt’s immer einen Konflikt zwischen Qualität und kurzer Entwicklungsdauer. Es gibt auch sog. Bananenprodukte, welche dann beim Kunden reifen.

Zudem sollte die Marktpräsenz/Einsatzdauer möglichst lange sein (Cash Cows BWL)

Wie verkürzt man die Entwicklungszeit?

* Motivation der Mitarbeiter, eingeschworenes Team
* Einsatz Spezialisten und mehr Mitarbeiter
* Vorsicht: Der "Chinesische Dreisatz" gilt nur sehr begrenzt.

Entwicklungsdauer = LOC / (P \* M)

LOC = Lines ofCode

P = Produktivität (LOC /Mannjahr)

M = Personalkapazität (Anzahl Mitarbeiter)

Ertrag

Umsatz und Kosteneinsparung:

Softwarehaus (Lieferant/Produzent)

* Lizenzeinnahmen ( Verkaufszahl \* Preis)
* Lizenzgebühr für das Nutzungsrecht
* Wartungsgebühr
* Beratung

Kunde/Anwender

* Kosteneinsparungen durch Rationalisierung (was braucht man wirklich? Unnötiges weglassen)
* Amortisation der Anschaffungskosten
* Nicht quantifizierbare Komponenten(Schnellere Reaktionszeit (Anfrage... Lieferung), Bessere Information)

1.3 Warum braucht man SE?

* Zunehmende Qualitätsanfoderungen
* Zunehmend Außer-Haus-Entwicklung (abgeben von Projekten an externe Firmen)
* Zunehmende Altlasten (manche Systeme sind bis >=20 Jahre im Einsatz!)

SE wird in der Entwicklung, Betrieb und Pflege verwendet.

Ziel im SE ist es qualitativ hochwertige Software zu bauen. Dafür gibt’s Prinzipien, Methoden, Vorgehensweisen, Werkzeuge..