universitätbonn ROOTS

Kapitel 1

Software Engineering – Überblick

– Stand: 14.10.2014 –

Ziele

Kosten

- Softwarekosten übersteigen oft Systemkosten
  - Beispiel: PC
- Softwarewartung übersteigt Entwicklungskosten
  - Bis zu 80% der Gesamtkosten

Qualität

- Schlechte Software schadet
  - Direkte Schäden
  - Unzufriedene Benutzer
  - …

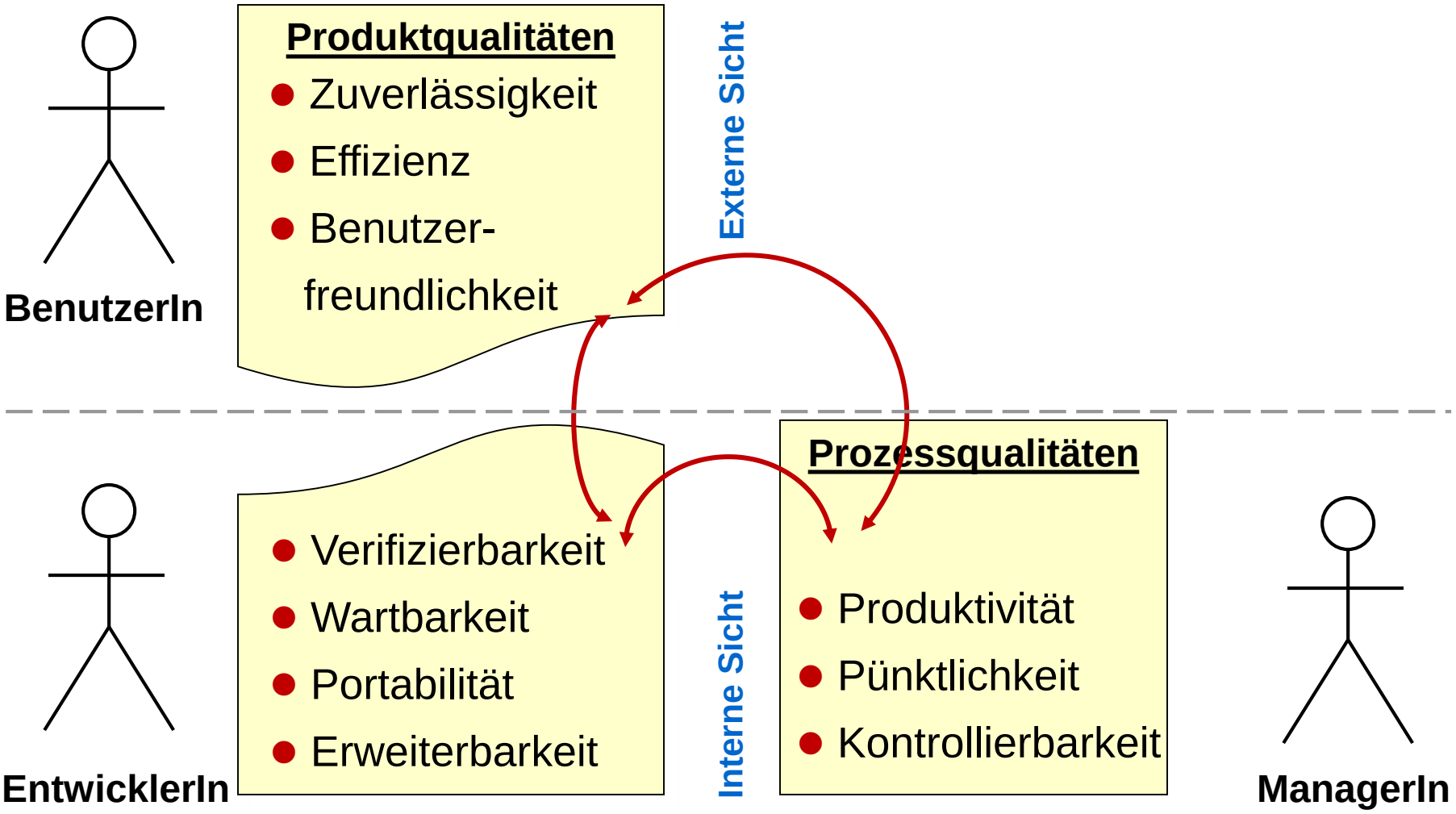
→ Kosteneffektive Entwicklung von qualitativ hochwertiger Software

Fehlleistungen (Beispiele)

- Zu „enger“ Weltausschnitt
  - Im Jahre 1992 erhielt Mary (geb. 1888) eine Einladung in den Kindergarten
- Fehlverhalten
  - Gepäcksystem am Flughafen von Denver
  - beschädigte Koffer
  - 3,2 Mio. $ über dem Budget
  - 16 Monate verspäteter Start mit zum größten Teil manuellem System
- Unnötig komplexe Lösung
  - C-17 Transportflugzeug ▸ 19 Computer, 6 Programmiersprachen
  - 500 Mio $ über dem Budget
- Interface-Missbrauch
  - Wissend, dass es ein System gab, das die Abfahrt des Zuges mit offenen Türen verhinderte, fixierte ein Zugführer den Startknopf mit Klebeband in Startstellung.
  - Als er den Zug verließ, um eine klemmende Tür zu schließen, fuhr der Zug ohne ihn los (als die Tür nicht mehr klemmte).

Er fand dies eine besonders clevere Art sicherzustellen, dass der Zug sofort losfuhr sobald die letzte Tür geschlossen war.

Software-Qualitäten: Verschiedene Sichten



Herausforderungen

Komplexität

- Viele Funktionen
- Viele Ziele ▸ Interessenskonflikte
- Viele Komponenten ▸ Komplexität der Zusammenstellung
- Viele Teilnehmer ▸ Komplexität der Kommunikation

Keine Einzelperson kann ein ganzes System verstehen

Änderung

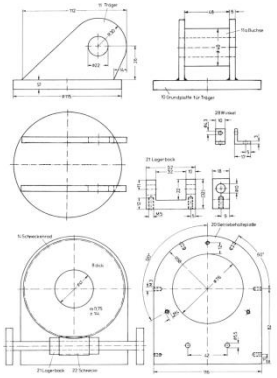
- Welt ▸ Gesetzgebung, Geschäftsabläufe, ...)
- Der modellierte Ausschnitt ▸ mehr Funktionalität, Systeme integrieren, ...
- Die Implementierungstechnologie ▸ neue Sprachen, Komponenten, ...
- Das Team ▸ Personen gehen, neue kommen hinzu, ...
- Das Management ▸ neue Geschäftsausrichtung, ...

Software ist ständiger Änderung unterworfen!

Andere Ingenieurswissenschaften

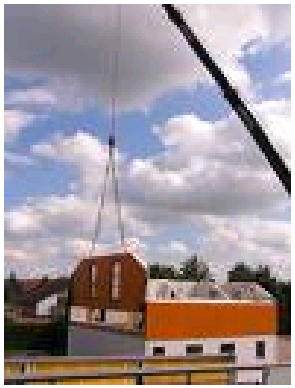
Domäne

- Klar definierte Probleme
- Ein Produkt muss *gebaut* werden
- Hohe Qualitätsanforderungen



Methoden

- Systematische Verfahren und ihre disziplinierte Anwendung
- Standardschnittstellen, -komponenten und -prozesse
- Wissen um verfügbare Komponenten
- Empirische Methoden zur Bewertung von Lösungen



Software Engineering ist anders

- Andere Ingenieursbereiche
  - Herstellung bestimmt die Endkosten
  - Änderungen sind ziemlich selten
    - ⇨ 2000 Jahre von der Euklidischen zur nicht-Euklidischen Geometrie
  - Änderungen sind sehr teuer
    - ⇨ Redesign wird gründlich durchdacht
    - ⇨ Auswirkungen werden genau analysiert
- Software Engineering
  - Herstellung ist eine einfache Duplizierung
  - Änderungen geschehen dauernd
    - ⇨ Manchmal innerhalb von Stunden (Kunde hat seine Meinung geändert)
  - Änderungen sind einfach
    - ⇨ Kein durchdachtes Redesign
    - ⇨ Auswirkungen werden nicht ausreichend bedacht

Andauernde Änderungswünsche zusammen mit der Leichtigkeit, Änderungen durchzuführen, führen zu ungenügend überdachten, ad-hoc „Lösungen“.

Software Engineering

Professionelle Softwareentwicklung

- Werkzeuge
  - Konzepte, Sprachen, Systeme, …
- Methoden
  - Wie setzen wir die Werkzeuge ein?
- Prozesse
  - Wie organisieren wir das ganze?

Übersicht: Themenbereiche

Teamarbeit und Änderungen
- Configuration Management
- Issue Management
- Rationale Management
- Project Management

Objektorientierte Modellierung
- OOP → UML
- Design Patterns
- Refactoring

SE Aktivitäten („Workflows“)
- Anforderungserhebung
- Analyse
- Systemdesign
- Objektdesign
- Testen

Softwareprozess
- Wasserfall-Modell
- V-Modell
- Spiral-Modell
- Unified Process
- Agile SW-Entwicklung