

Software Engineering

Mark Keinhörster

FOM

Hochschule für Ökonomie und Management

23. Februar 2017

1 Vorraussetzungen

2 Einführung

3 Vorgehensmodelle

4 Requirements Engineering

5 SW Architektur

6 Real-Time SE

7 Projektmanagement

Vorraussetzungen

Was möchten Sie gerne behandeln?



Was sollten Sie am Ende können?

- Software Engineering als Teildisziplin der Informatik kennen
- Grundpfeiler des Software Engineering kennen
- Vorgehensmodelle der Softwareentwicklung beschreiben und abgrenzen
- Softwarequalität messen und bewerten
- Weiterführende Konzepte verstehen und anwenden

Vorraussetzungen

Einführung

Vorgehensmodelle

Monumental

Agil

Requirements
Engineering

SW Architektur

Real-Time SE

Projektmanagement

Projektplanung

Qualitätssicherung

Messen/Bewerten

Einführung

Aufgabe: Rekursives Take

Implementieren Sie die Methode `take(int n)` die die ersten `n` Elemente eines Übergebenen Arrays vom Typ `int` als neues Array zurückgibt.



- Anforderungen mehrere 100 Seiten lang
- Anforderungen unklar, widersprüchlich, flexibel
- International verteilte Teams
- Mehrere tausend Nutzer in 5 Ländern
- Unterstützung von Chrome, Firefox, IE 6
- 6 Monate Projektlaufzeit, 500.000 LOC



Standish Group (<http://www.standishgroup.com>)
veröffentlicht jährlich "Chaos Report".

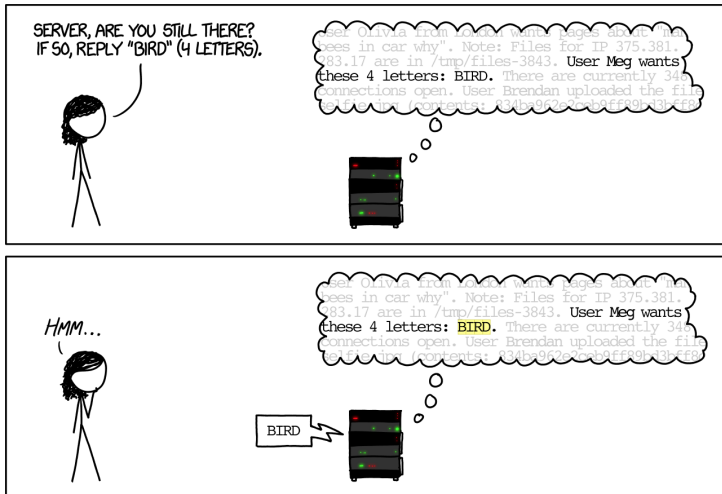
Chaos Report 2015

- 19% der betrachteten IT-Projekte scheitern
- 52% der betrachteten IT-Projekte drohen zu scheitern
- 29% der betrachteten IT-Projekte sind erfolgreich

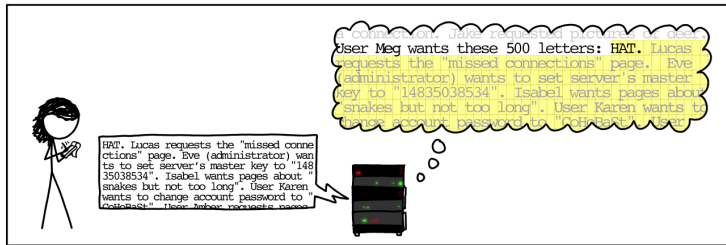
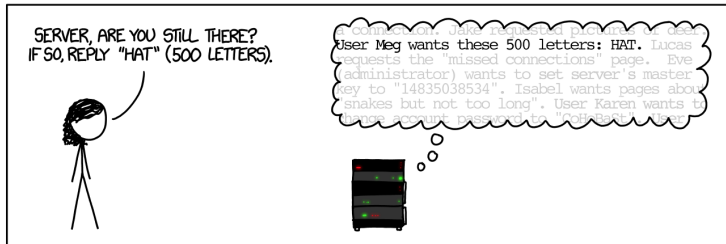
On June 4, 1996, on its maiden flight, the Ariane-5 was launched and performed perfectly for approximately 40 seconds. Then it began to veer off course. At the direction of the Ariane ground controller, the rocket was destroyed by remote control. . . . total cost of the disaster was 500 million dollar.

- Flugbahn wird durch “Inertial Reference System (SRI)” gemessen, dessen Software teilweise von Ariane-4 übernommen wurde.
- Andere Flugbahndaten erzeugten Überlauf bei Konvertierung von 64-Bit Floating Point in 16-Bit Integer und verursachten Fehlfunktion des SRI-Systems.

- Heartbeat hält TLS-Verbindung am Leben
- Eine Seite schickt beliebig langen Payload, Gegenseite schickt die gleichen Daten wieder zurück
- Indikator für aufrechte Verbindung



- Prüfung ob Payload der angegebenen Länge entspricht fehlte
- War der Payload kürzer als angegeben wurden Daten aus den darauffolgenden Speicherbereichen kopiert
- Da OpenSSL eine eigenen Speicherverwaltung implementiert waren diese Daten auch aus dem OpenSSL Kontext



SW-Entwicklung in den 40er und 50er Jahren

- Teure Hardware
- Low-Level Programmierung (Assembler, fast kein OS)
- Von Experten bedient (Entwickler = Nutzer)
- numerisch-naturwissenschaftliche Probleme
- Codierung bekannter, mathematisch fundierter Algorithmen
- Viele Daten, einfache Algorithmen
- Häufig Batch-Systeme
- Fokus auf Effizienz
- Häufig "Wegwerf-Software"

SW-Entwicklung in den 60er Jahren bis heute

- Preiswerte Hardware mit viel Leistung
- Embedded Hardware die günstig ist und häufig eingesetzt wird
- Nicht-Informatiker
- Vielzahl von Anwendungsbereichen
- Kritische Anwendungsbereiche wie Finanzsektor etc.
- Systeme sind komplex und interaktiv
- Software teurer als Hardware
- Lange Lebensdauer

Softwarekrise

- Programme werden immer komplexer
- Passende Programmiersprachen, Methoden, Werkzeuge fehlen

Folgen

- Kosten für Software steigen
- Softwareprojekte scheitern

Lösungsansatz

SW-Entwicklung als Ingenieurstätigkeit mit definiertem Vorgehen statt künstlerischer Tätigkeit

Software
Engineering

Mark Kein Hörster

Vorraussetzungen

Einführung

Vorgehensmodelle

Monumental

Agil

Requirements
Engineering

SW Architektur

Real-Time SE

Projektmanagement

Projektplanung

Qualitätssicherung

Messen/Bewerten

Dijkstra (The Humble Programmer)

Als es noch keine Rechner gab, war auch das Programmieren noch kein Problem, als es ein paar leistungsschwache Rechner gab, war das Programmieren ein kleines Problem und nun, wo wir gigantische Rechner haben, ist 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.

IEEE Definition

Software ist eine Sammlung von Computerprogrammen, Prozeduren, Regeln, zugehöriger Dokumentation und Daten

- Programme sind eine Teilmenge von Software
- SW beinhaltet Dokumente die verschiedene Abstraktionsschichten für verschiedene Zielgruppen beschreiben

- Kommunikationsprobleme mit Anwender
- SW ist immateriell
- SW ist leicht modifizierbar, Behebung von Fehlern wird unterschätzt
- SW ist nur beobachtbar
- Anforderungen ändern sich regelmäßig
- SW altert ohne zu Verschleiß, das führt zu immer neuen Erweiterungen und wachsender Komplexität
- Verhalten für Software lässt sich nur schwer beweisen
- ...

IEEE Definition

Software Engineering der systematische Ansatz für

- die Entwicklung,
- den Betrieb
- sowie die Wartung

von Software.

Effiziente Entwicklung von messbar qualitativ hochwertiger Software

- Korrektheit und Zuverlässigkeit
- Robustheit
- Effizienz
- Benutzerfreundlichkeit
- Wartbarkeit und Wiederverwendbarkeit

Qualitätsfaktoren

- Extern (für den Benutzer sichtbar)
- Intern (nur für den Entwickler sichtbar)

Vorgehensmodelle

Monumentale Vorgehensmodelle

Agile Vorgehensmodelle

Requirements Engineering

Software Architektur

Real-Time Softwareengineering

Projektmanagement

Projektplanung

Qualitäts-sicherung

Messen und Bewerten