Software Engineering

Mark Keinhörster

FOM
Hochschule für Ökonomie und Management

26. Februar 2017

Inhaltsverzeichnis

Vorraussetzungen

Ihre Erwartungen an die Veranstaltung

Was möchten Sie gerne behandeln?

-
-
-
-
-

Erwartungen an die Veranstaltung

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

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.

Die Realität

- 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

Softwarekatastrophe ARIANE 5

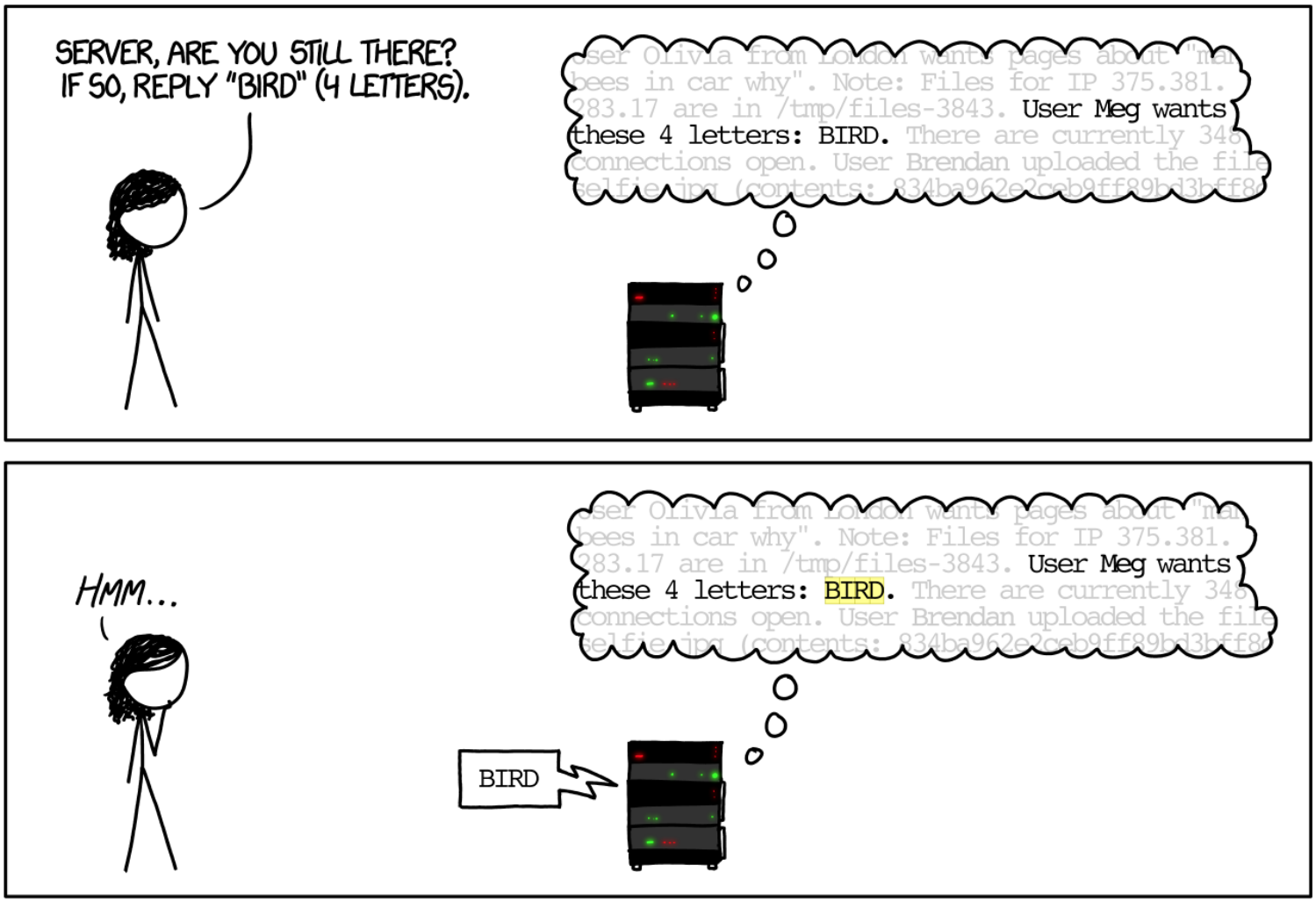
On June 4, 1996, on its maiden flight, the Ariane-5 was launched and performed perfectly for approximately 40 seconds. Then it began to veer off course. At the direction of the Ariane ground controller, the rocket was destroyed by remote control. . . . total cost of the disaster was 500 million dollar.

- 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.

Softwarekatastrophe Heartbleed

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

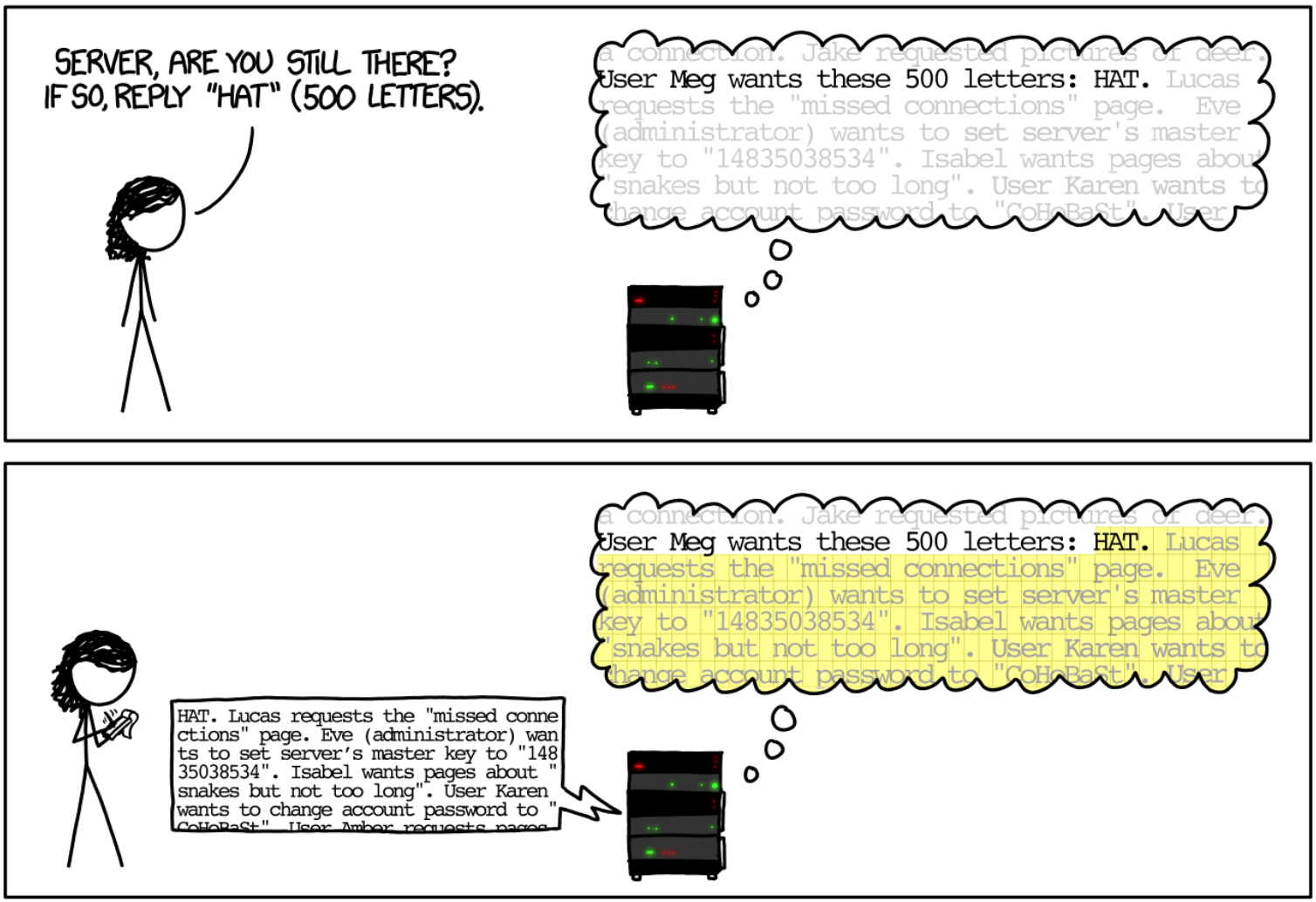
Softwarekatastrophe Heartbleed



Softwarekatastrophe Heartbleed

- 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

Softwarekatastrophe Heartbleed



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 nutzen die Software
- 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

Djikstra (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 beeinhaltet Dokumente die verschiedene Abstraktionsschichten für verschiedene Zielgruppen beschreiben

Probleme bei der Softwareentwicklung

- 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 über Umgebung ohne zu Verschleißen, das führt zu immer neuen Erweiterungen und wachsender Komplexität
- Verhalten für Software lässt sich nur schwer beweisen
- ...

- Auslöser für Begriff "Software Engineering" war Softwarekrise von 1968
- Begriff "Software Engineering" wurde 1967 von F.L. Bauer (ehemaliger Prof. in München) im Rahmen einer "Study Group on Computer Science" der NATO geprägt.
- Software wurde erstmals als Industrieprodukt bezeichnet

Definition IEEE

Software Engineering ist der systematische Ansatz für

- die Entwicklung,
- den Betrieb
- sowie die Wartung

von Software.

Definition Lehrbuch Software-Technick (Balzert)

Zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden, Konzepten, Notationen und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Software-Systemen. Zielorientiert bedeutet die Berücksichtigung z.B. von Kosten, Zeit, Qualität.

Ziele des Software Engineerings

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)

Software Engineering als Prozess

Der systematische Ansatz im Software Engineering wird auch als Entwicklungsprozess bezeichnet. Er beeinhaltet eine Reihe von Aktivitäten die zur Entwicklung von Software führen.

- Software spezifizieren
- Software entwickeln
- Software validieren
- Software weiterentwickeln

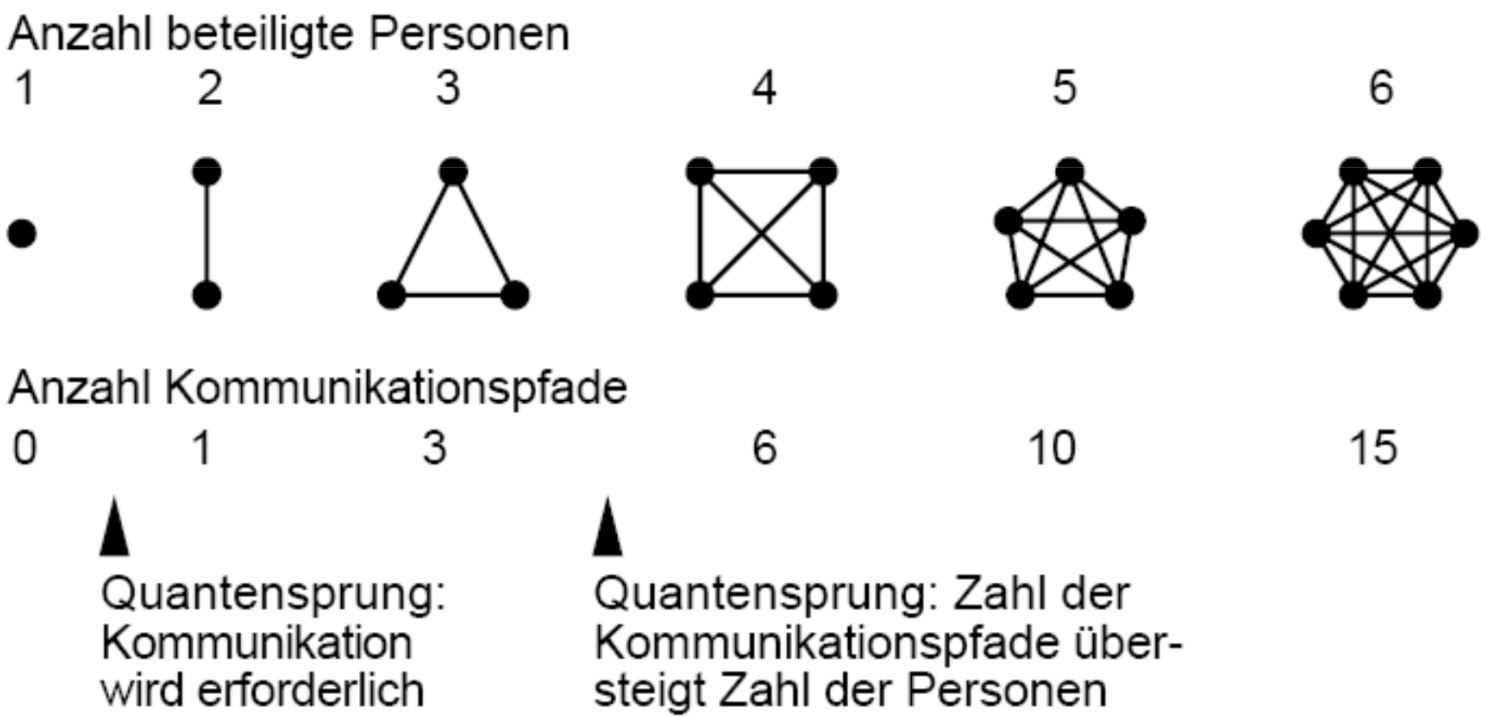
Kleine Softwareprojekte

- Wenige LOC
- SW für die eigene Verwendung
- Produkt spezifiziert sich selbst
- Lösung wird direkt entwickelt
- Validierung und Korrekturen am Endprodukt
- 1 Entwickler
- Komplexität gering
- Software besteht aus wenigen Komponenten
- Wenig bis keine Dokumentation nötig
- Keine Planung und Projektstruktur nötig

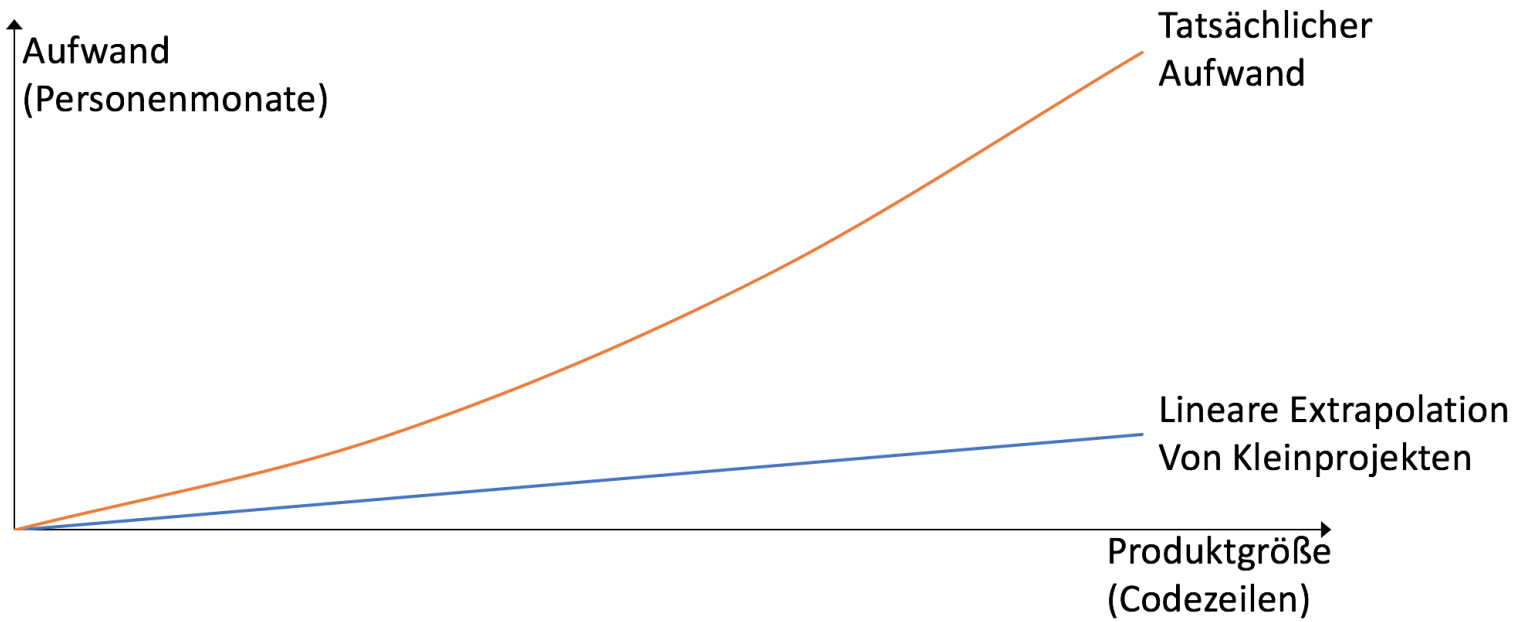
Große Softwareprojekte

- Viele LOC
- SW für die Verwendung durch Dritte
- Klares Ziel, genaue Spezifikation erforderlich
- Lösung wird in Phasen entwickelt
- Tests in jeder Phase sind unerlässlich
- Produkt wird im Team entwickelt
- Hohe Komplexität macht Strukturierung der SW erforderlich
- Software besteht aus vielen Komponenten
- Dokumentation für den wirtschaftlichen Betrieb der SW erfoderlich
- Projektstruktur zwingend erforderlich

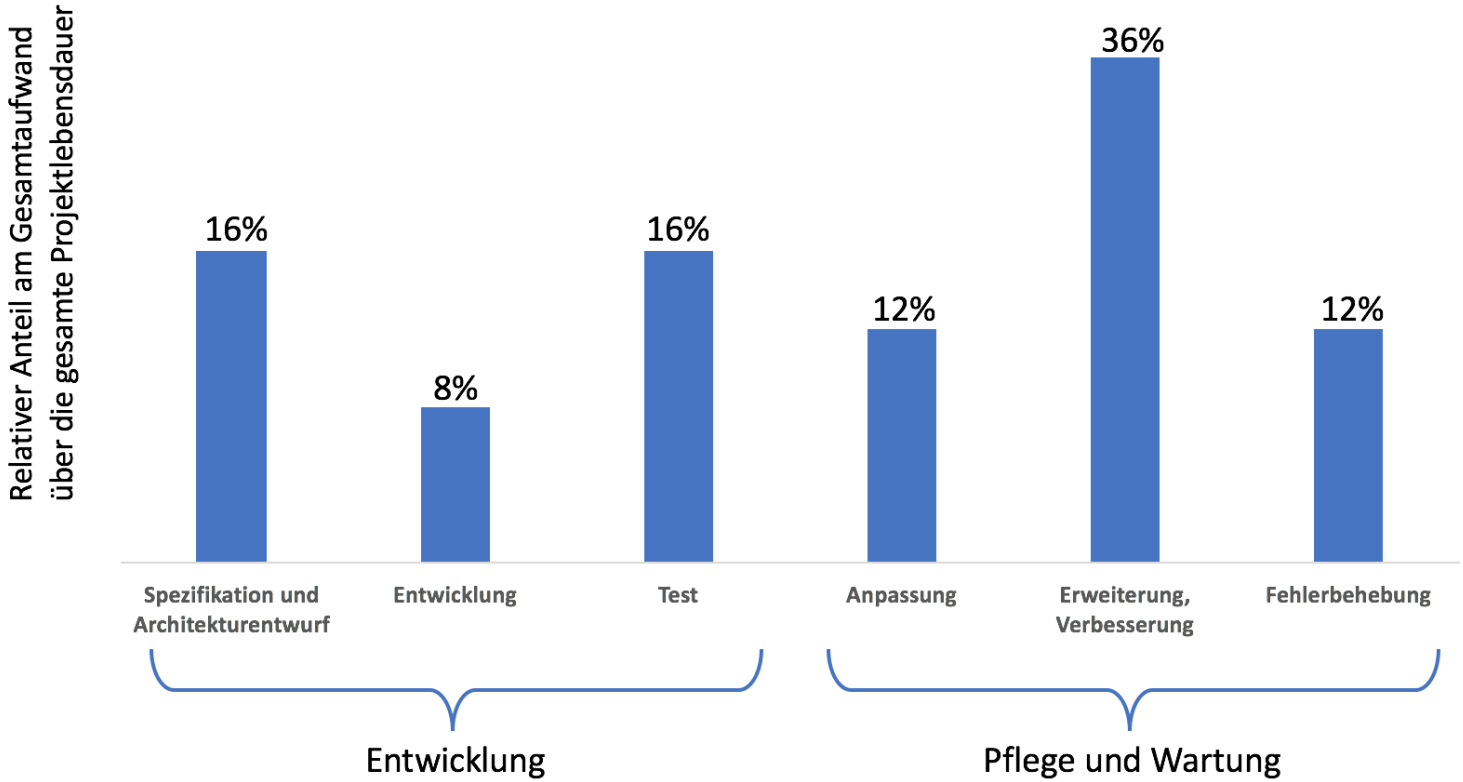
Wachstum des Kommunikationsbedarfs



Wachstum des Aufwands



Aufwände aufgeschlüsselt



Übung 1.1

Eine Person braucht zum Bau einer 2m langen Bruecke 0,5 Tage. Wie lange brauchen 100 Leute für den Bau einer 2km langen Brücke?

1. Interpolieren Sie den Aufwand linear
2. Warum ist die Berechnung aus Punkt 1 eine Milchmädchenrechnung?

Übung 1.1 - Lösung

Eine Person braucht zum Bau einer 2m langen Bruecke 0,5 Tage. Wie lange brauchen 100 Leute für den Bau einer 2km langen Brücke?

1. Aufwand = (2000m / 2m * 0.5PT) / 100 Personen = 5 Tage
2. Mehr Kommunikation, Projekt deutlich Komplexer, Ressourcenbeschaffung, Logistik ...

Übung 1.2

Eine Kundenbetreuerin im Firmenkundengeschäft einer Bank hat auf Grundlage eines Tabellenkalkulationsprogramms eine kleine persönliche Anwendung geschrieben, die sie bei der Überprüfung der Kredite der von ihr betreuten Firmen unterstützt. Die notwendigen Daten gibt sie jeweils von Hand ein. Der Abteilungsleiter sieht diese Anwendung zufällig, ist davon angetan und beschließt, sie allen Kundenbetreuerinnen und -betreuer zur Verfügung stellen. Die notwendigen Daten sollen jetzt automatisch aus den Datenbanken der Bank übernommen werden. Die Kundenbetreuerin gibt an, für die Entwicklung ihrer Anwendung insgesamt etwa vier Arbeitstage aufgewendet zu haben. Der Abteilungsleiter veranschlagt daher für die Übernahme und die gewünschten Änderungen einen Aufwand von einer Arbeitswoche. Als die geänderte Anwendung endlich zur Zufriedenheit aller Beteiligten läuft, sind jedoch rund acht Arbeitswochen Aufwand investiert. Der Abteilungsleiter erzählt die Geschichte einem befreundeten Berater als Beispiel, dass Informatikprojekte nie ihre Termine einhalten. Darauf meint der Berater trocken, der investierte Aufwand sei völlig realistisch und normal. Begründen Sie warum.

Übung 1.2 - Lösung

- Die Kundenbetreuerin hat das Fachkonzept ihrer Tabellenkalkulationsanwendung vermutlich schon vor den angegebenen vier Tagen Bearbeitungszeit im Kopf gehabt. Die Fremdentwickler müssen dieses zumindest erst nachvollziehen.
- Die neue Anwendung ist durch die Datenbankanbindung mit den entsprechenden Schnittstellen und Zugriffsrechteproblematiken deutlich komplexer.
- Der Kommunikationsaufwand schon allein von Kundenseite (viele Berater = viele unterschiedliche Meinungen) ist erheblich
- Die neu entstandene "professionelle" Anwendung hat einen erheblich höheren Aufwand für die Validierung als eine eigengenutzte Entwicklung.
- An die Bedienbarkeit (Nutzerschnittstelle) werden bei einer "professionellen" Anwendung erheblich höhere Ansprüche gestellt.

Vorgehensmodelle

Monumentale Vorgehensmodelle

Agile Vorgehensmodelle

Projektmanagement

Projektplanung

Qualitaetssicherung

Messen und Bewerten

Requirements Engineering

Software Architektur

Konfigurationsmanagement