




Regensburg, 22. Mai 2017

Moderne Software-Technik

## **Gastvortrag – Software Renovierung**

Dr. Markus Pizka


# ... offensichtlich kein gewöhnliches Thema



Software Renovierung

Alle Bilder News Shopping Videos Mehr ▾ Suchoptionen

Ungefähr 530.000 Ergebnisse (0,35 Sekunden)



**Hinweis zum Datenschutz bei Google**

ICH MÖCHTE DAS SPÄTER LESEN **ANSEHEN**

**Renovierungsarbeiten - arbeitshilfe2000.de**

Anzeige [www.arbeitshilfe2000.de/](http://www.arbeitshilfe2000.de/) ▾

von anerck. gemeinnützigem Verein. Rufen Sie an, Tel. Augsburg 3494878  
Hilfe für Arbeitslose - AIDS Hilfe, gemeinnützig - Gesundheitsamt Augsburg

**Suchergebnis auf Amazon.de für: Renovierung: Software**

[www.amazon.de/Renovierung-Software/s?...n%3A301927%2Ck%3ARe...](http://www.amazon.de/Renovierung-Software/s?...n%3A301927%2Ck%3ARe...) ▾

Ergebnissen 1 - 12 von 49 - Bauherren, Architekten, Renovierung, Garten, Makler, Anbau, Bungalow. 2008 ... RTL 3D Software 3.0 - Einsatz in 4 Wänden. PC.

**Software hilft bei der Haus-Renovierung - PC-WELT**

[www.pcwelt.de](http://www.pcwelt.de) > Computer & Technik ▾

02.09.2004 - Ein Programm von Buhl soll Kosten und Ärger bei Renovierungsmaßnahmen vermeiden helfen.

**renovierung - Free Software download**

[www.freeware-download.com/tags/renovierung.html](http://www.freeware-download.com/tags/renovierung.html) ▾

Software Tag: renovierung. Alle Programme, mit dem Tag renovierung: 1 ... Tags: bau, haus, architektur, dekoration, renovierung, entwicklung, ...

Anzeigen

**Komplette Renovierungen**

[www.paul-renovierungen.de/](http://www.paul-renovierungen.de/) ▾

Renovierung und Sanierung günstig & komplett In München und Umgebung

**Renovierungen & Sanierung**

[www.renodal.de/](http://www.renodal.de/) ▾

Renovierung von Haus & Wohnungen, alles aus einer Hand in München

📍 München

☎ 089 38465278

**Renovieren von A-Z**

[www.ellmerboden.de/](http://www.ellmerboden.de/) ▾

Boden-Wand-Decke-Bäder  
Alles aus einer Hand

**Renovierungen in München**

[www.innenausbau-kuschel.de/](http://www.innenausbau-kuschel.de/) ▾

Renovieren, Sanieren und Innenausbau  
Wir machen Ihr Zuhause Kuschelig!

**Günstig renovieren?**


[www.blauarbeit.de/sanierung](http://www.blauarbeit.de/sanierung) ▾

Jetzt einen Profi für Ihre Modernisierung hier kostenlos finden!

Software Engineering

Alle Bilder News

Ungefähr 154.000.000 Ergebnisse



**Hinweis z**

ICH MÖCHTE D

**Softwaretechnik – Wi**

<https://de.wikipedia.org/wiki>

Die Softwaretechnik ist eine d  
Engineering und beschäftigt  
Aufwandsschätzung - Akzept

**Software engineering**

<https://en.wikipedia.org/wiki>

The result of the conference is  
[i.e., software engineering

**Was ist Software Eng**

<https://www.hs-heilbronn.de>

Software Engineering (B.Sc  
Software besteht die Herausfo

**TUM - Software Engi**

[portal.mytum.de](http://portal.mytum.de) > Sitemap

## Warum - Relevanz

- Umfang und Alter von IT-Systemen
- Phänomene

## Was - Handlungsalternativen

- Alternativen
- Strukturierte Bewertung

## Wie - Vorgehen

- Strategieentwicklung
- Besondere Aktivitäten und Anforderungen

## Beispiele

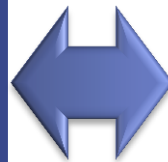
## ■ itestra GmbH

- Ursprung: Technische Universität München
  - Betriebssysteme, High Performance Computing
  - Software (Re-)Engineering, Qualität, Ökonomie
- Gründung 2004, 60 Mitarbeiter, 4+ Standorte
- **Leistung:** Strategieberatung und Engineering



München

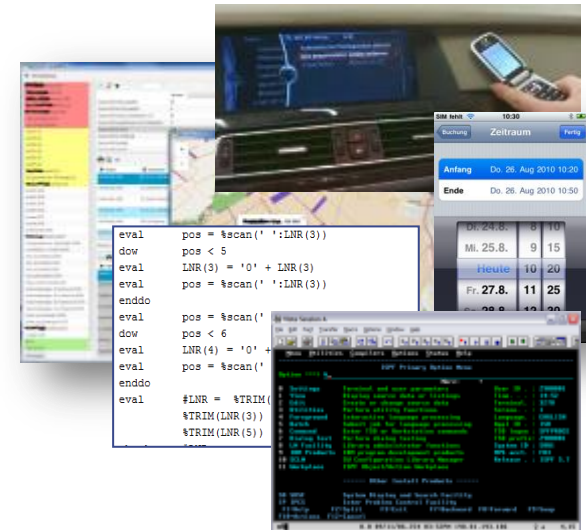
**Governance &  
Renovierung**  
„Altes fit machen“



**Business  
Solutions**  
„Neue Dinge“

## ■ Kunden:

- Primär namhafte Großunternehmen ~  
Versicherungen, Banken, Automobil, Handel
- In Summe ca. 100 Applikationen mit in Summe  
mind. 500 Mio LoC betrachtet.



	BUSINESS	MAINFRAME	MID & SMALL	S.PURPOSE	Web & Mobile
Programmiersprachen	Java, C#, C++, PL/SQL, ABAP	COBOL, PL/I, ASM, NATURAL, RPG, COOLGEN, SQL	RPG, VB/A Python, PHP, Tcl/Tk, Gupta	C, Ada, Lisp, MATLAB, Pearl	ObjectiveC, Java, JavaScript
Frameworks Bibliotheken	Java EE, .net, SAP ERP (MM, SD, PP(-PI), LE, FI/CO)		Typo3, vTiger, openCMS/ERP	MP/PVM, SOLR, JavaFX	Struts GWT, PrimeFaces, MyFaces AngularJS, NodeJS, BackboneJS, EmberJS ReactJS, Apache Cordova
OR-Mapping	Hibernate, TopLink				
Datenbanken	Oracle, DB2, PostgreSQL, SQL Server	DB2, IMS/DB, VSAM	MySQL, Access, MariaDB	SQLite	HTML5 Storage, MongoDB
Plattformen	Linux, Solaris, Windows SAP Web-AS 6, SAP NetWeaver7	MQ CICS, IMS/DC z/OS, Linux	Windows, Linux	LynxOS, Windows RT	Android, iOS, Windows Mobile

SchnittstellenMQS, SOAP, REST, JSON, CTG, SAP NPI, Spring Integration

## SOFTWARE ENGINEERING

Unified Process, Scrum, V-Modell, Use Cases, OO Design, UML, Inspektionen, CM, KM, etc.

# Aus Legacy wird Cutting Edge

## Einladung zum itestra Student Day

- **Planung & Projektrealität** - Halbzeitanalyse einer Migration nach Java und Eclipse RCP
- **Aus Legacy wird Cutting Edge** - Europas schnellste Schadensabwicklung made by itestra
- Ausklang des Tages bei einem gemeinsamen Essen



Freitag, **30.06.2017**, 15.00-19.00 Uhr

itestra GmbH, Destouchesstraße 68, 80796 München

Die Fahrtkosten in Höhe einer Bahnfahrt 2. Klasse trägt die itestra GmbH.

Melde dich bis zum **25.06.2017** an.

Sei mit dabei!

Anne-Kristin Hauk – [hauk@itestra.de](mailto:hauk@itestra.de)

## Bringen Sie Informatik zur Wirkung!

Ein erheblicher Teil der Informatiker arbeitet im Beratungsumfeld. In der Beratung lösen Sie kontinuierlich neue Fragestellungen bei verschiedenen Kunden und erlangen in den Projekten breites Wissen. Erfahren Sie aus erster Hand, welche spannenden Möglichkeiten Software-Beratung bietet, und probieren Sie aus, ob dieses Berufsfeld zu Ihnen passt!

### Inhalte des Workshops:

- Wir diskutieren mit Ihnen, was ein Software Consultant genau macht und warum es sich lohnt, Berater zu sein.
- Sie bearbeiten im Team eine anspruchsvolle IT-Fallaufgabe im Rahmen eines realen Software-Migrationsprojektes unter Berücksichtigung der technischen, ökonomischen und organisatorischen Rahmenbedingungen. Bei der Lösung unterstützen Sie unsere erfahrenen Kollegen.

Donnerstag, **06.07.2017**, 10.00-17.00 Uhr

itestra GmbH, Destouchesstraße 68, 80796 München

Melden Sie sich unter Angabe Ihres Semesters bis zum **26.06.2017** an.  
Wir freuen uns auf Sie!

Kontakt: Anne-Kristin Hauk ([hauk@itestra.de](mailto:hauk@itestra.de)) – [www.itestra.com](http://www.itestra.com)

# RELEVANZ



# Buzzword „Digitalisierung“ – Ihre Chance



- Radikale Veränderung von Produkten und Geschäftsprozessen
- Auf Grundlage der technischen Vernetzung und des Informationsaustausch zwischen Systemen, Geräten, Menschen.  
➔ vernetzt, geteilt, mobil
- Jede Information ist jederzeit, ortsunabhängig, ohne Verzögerung, nahezu kostenlos verfügbar, änderbar, generierbar.

*Möglich durch: Rechenleistung, Speicherkapazitäten, schnelle (Funk)netze, mobile Geräte, Sensoren, wearables, IoT, AI, ...*

- Hohe Änderungsgeschwindigkeit, hohe Reichweite (PoS!), anonym/keine Markenloyalität, **rasche Monopolbildungen**



# Die Wirkungen sind beobachtbar ...



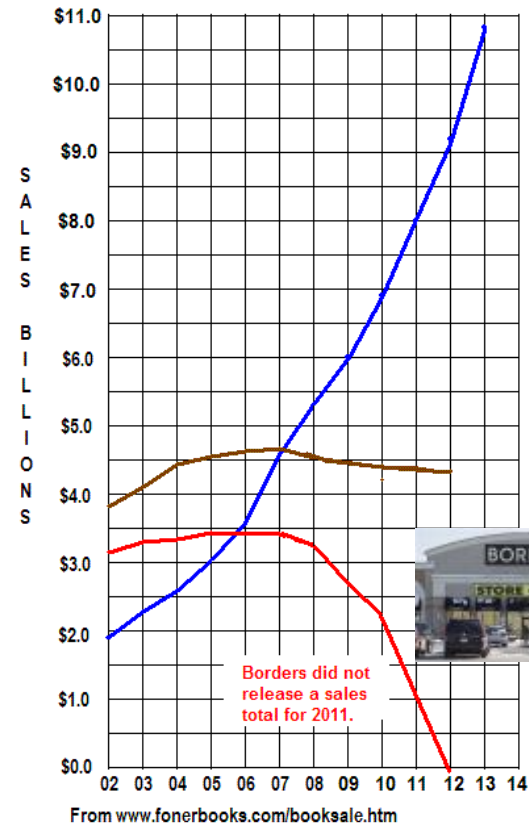
Wikipedia: „früher einer der weltweit bedeutendsten Hersteller für fotografische Ausrüstung“

EK (Eastman Kodak Co.) NYSE + BATS  
12-Oct-2011 9:34am Op 1.42 Hi 1.42 Lo 1.41 Last 1.41 Vol 8,700 Chg +0.00 (+0.00%) —



Obwohl Kodak Pionier der digitalen Fotografie war (schon 1975)!

Amazon North America Media Sales —  
Barnes & Noble Chain Store Sales —  
Borders Chain Store Sales —



Borders did not release a sales total for 2011.

Unter Top-100 Unternehmen heute: Apple (1976/97), Google (1998), Facebook (2004), Amazon (1994)

# Und erfasst weitere Branchen ... auch uns



Print, Versicherungen, Banken, TV, Taxi, ...

Software, Gesundheitswesen, Justiz, ...

## #1 ZEIT

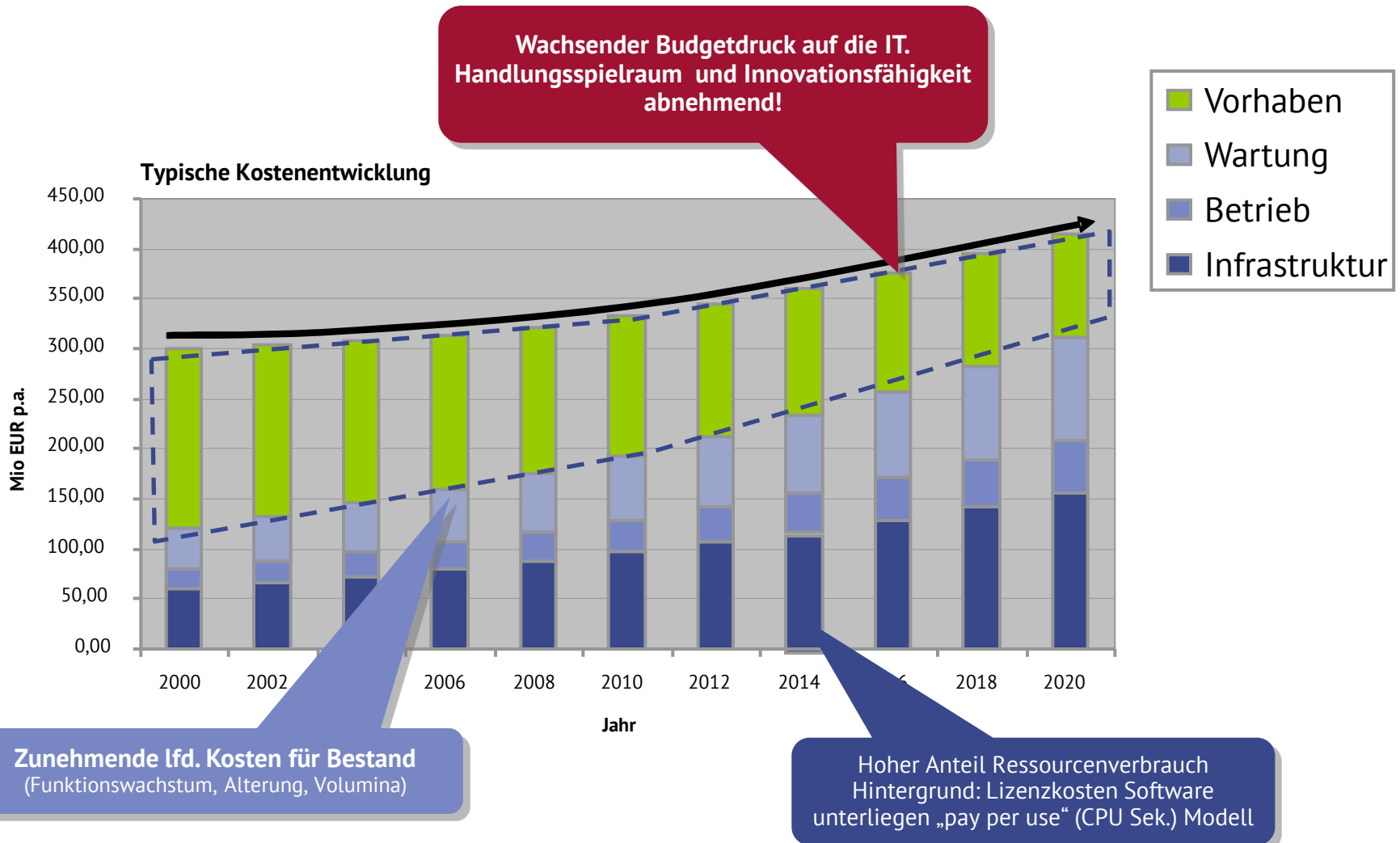
- Time-to-Market für neue Produkte und Dienste im Kerngeschäft
- Innovationsfähigkeit / Wettbewerbsvorteile

## #2 KOSTEN

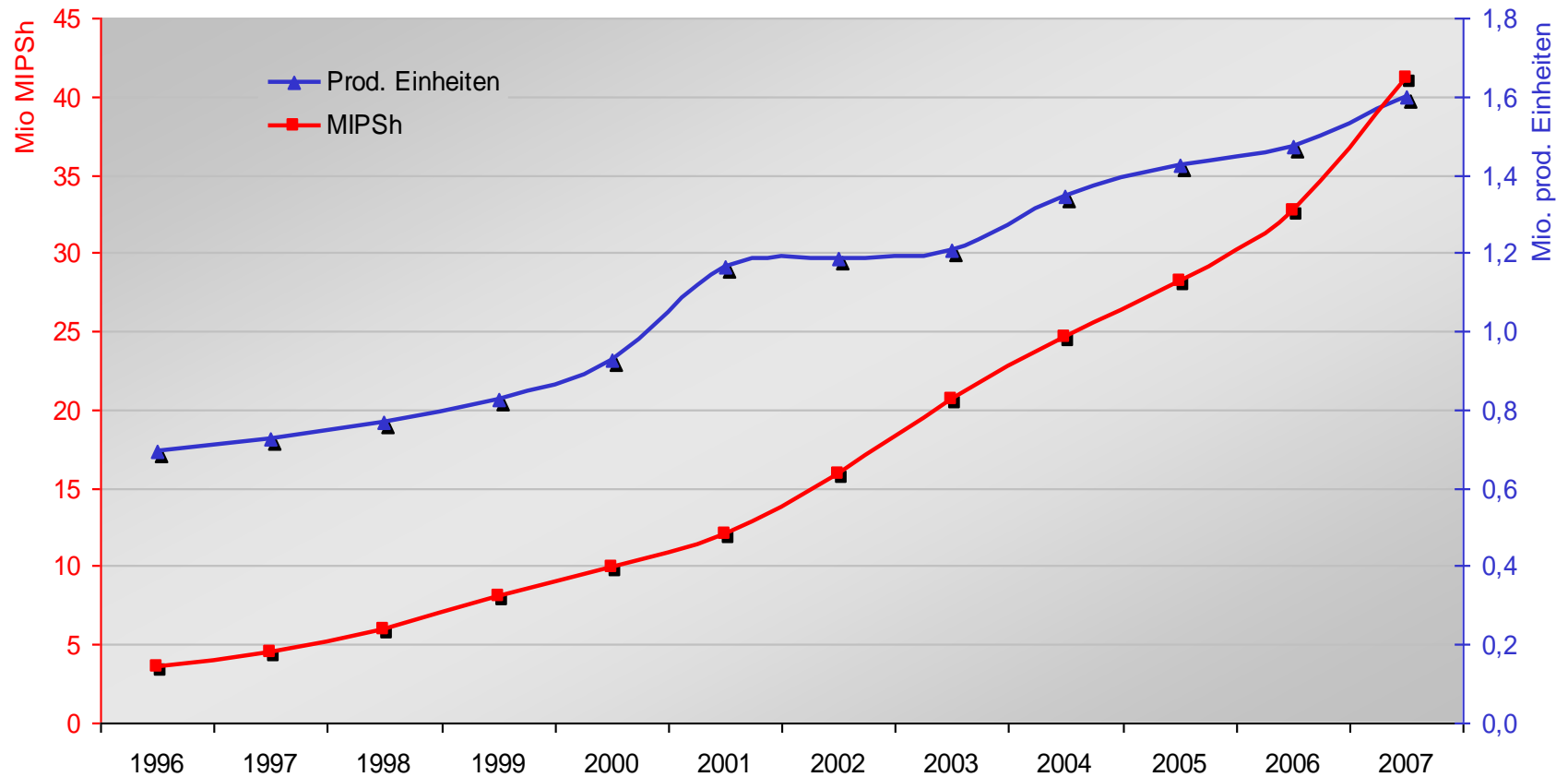
- Die Entwicklung und der Betrieb von IT-Systemen wird häufig als „teuer“ betrachtet (wenngleich i.d.R. nur geringer Anteil an den Unternehmenskosten).
- IT Budgets sind (stark) beschränkt und werden verschiedentlich weiter reduziert.

Software-Entwicklung ist teuer und langwierig!

# Realität: das verfügbare Budget ist knapp



# Trends, Bsp.: Industrie



**COBOL**

**MAINFRAME**

**JMS**

**VSAM**

**Host**



- Veraltet, Mängel, Workarounds, undocumented, ...
- PS: ... und 10 Jahre „junge“ Java Systeme sind teilweise auch schon „Legacy“

- Durchschnittliches Alter: ca. 25 Jahre (1967–2014)
- Umfang
  - **groß bis sehr groß**, d.h. 0,5 bis **10 MLoC** (bzw. 100 in AWL)
  - Kleine Systeme werden bei Bedarf ersetzt → hier nicht relevant
- Technologie: sehr oft alt (COBOL, PL/I, ASM, RPG, C++, VaGen, Synon, Easytrieve, PowerBuilder, ...) aber ggf. auch in C# oder Java ... mit alten Frameworks
- Erneuerungsbedarf
  - Software verschleißt nicht, **Software altert** relativ zu seiner Umgebung – **fachlich und technisch durch** Änderungen in der realen Welt und technischen Fortschritt.
  - **Fachlich:** Erfüllung der aktuellen Anforderungen (I-Net, mobile, Self-Service, ...)
  - **Technisch:** Umstieg auf unterstützte Technologie (Sprache, Framework, Werkzeuge, etc.)
- Auf Basis anderer/alter Annahmen realisiert (z.B. Speicher ist teuer)
- **Sind enorm wichtig und beinhalten enorme Mengen wertvoller Daten.**
- Qualitätsmängel





# COBOL

## Volumen:

**200 billion lines of COBOL** are in use today (65% of the total software). This represents a **\$2 trillion dollar total investment**. - Gartner Group

**5 billion lines of new COBOL** are developed **every year**. - Gartner Group.

COBOL applications **process over 80% of all daily business transactions** and mainframe platforms store 70% of all the data

Quelle: Future of COBOL, LegacyJ Corporation. 2003.  
<http://www.legacyj.com/cobol/FutureOfCobol.pdf>

**... und der Umfang  
wächst weiter.**

**Ihr Konto  
Ihre Versicherung  
...**

===>

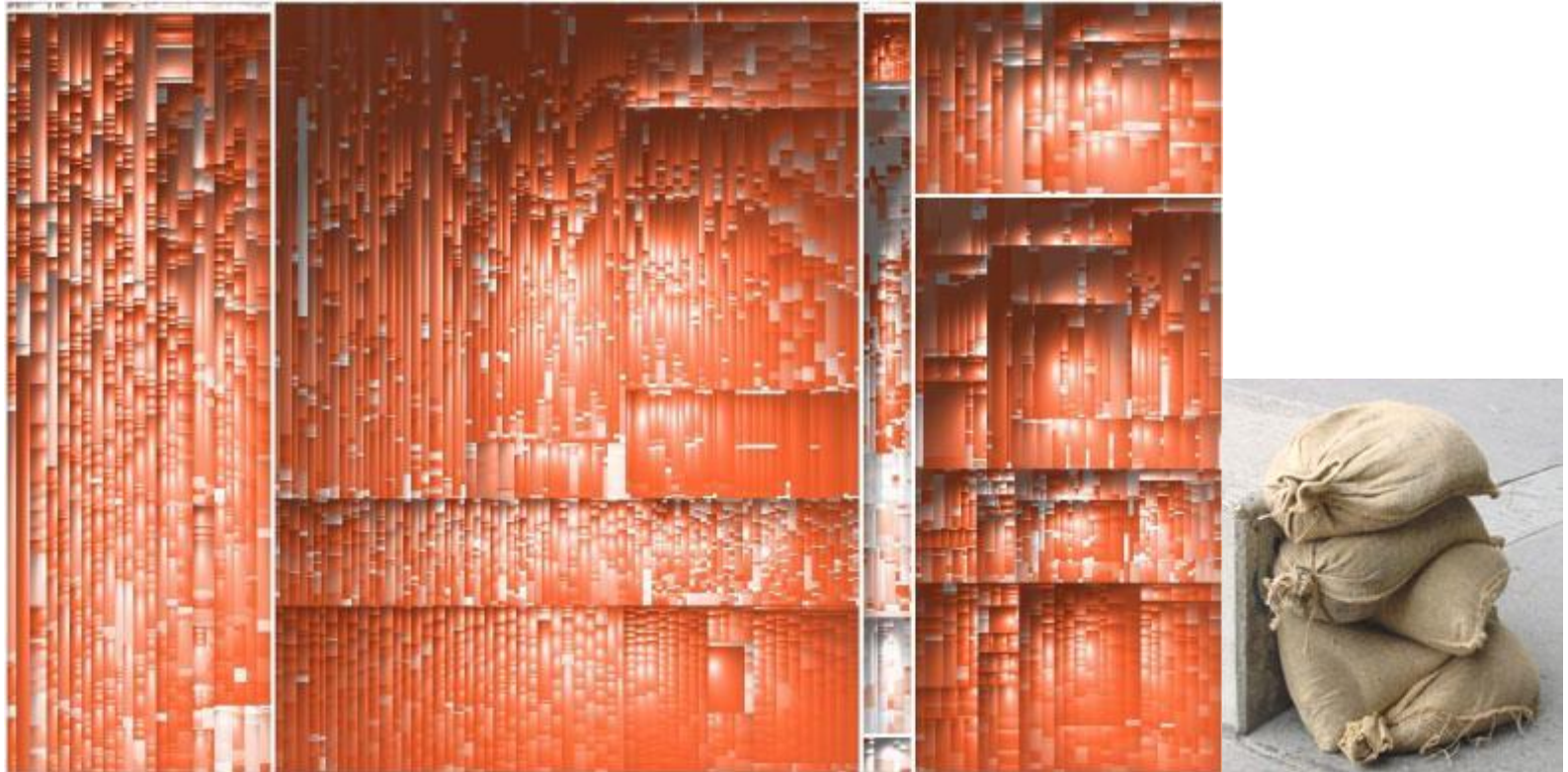
# Strukturverlust – S. Eick, Bell 2001

- 100 MLoC, 5000 Module, 15 Jahre, 10.000 Entwickler
- Verlust Separation of Concern
- Änderungsaufwand und Fehlerpotential steigen



# Zunehmender Ballast

Copy + Paste: (3% best), 50% average, 90% worst

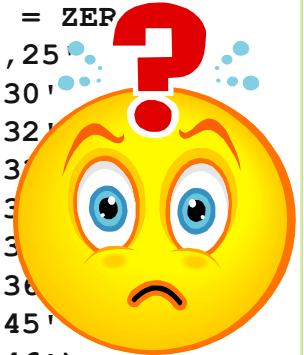


... zusätzlich nicht erreichbar („*dead code*“) häufig 30% + unused (?%) + ...

```
IF ZVDTA-BIS-CJMT (IDFA9) NN
OR ZNDFA-MT (IDFA9) NUMERIC
AND ZNDFA-MT (IDFA9) > 96
OR (ZNDFA-SA (IDFA9) = ,40'
OR ZNDFA-SA (IDFA9) = '69')
AND ZNDFA-VA (IDFA9) > 96
OR ZNDFA-MT (IDFA9) NUMERIC
AND ZNDFA-BIS-CJ (IDFA9) NOT = WA-VORJAHR-CJ
AND ZNDFA-BIS-CJ (IDFA9) NOT = WA-ABU-CJ
OR ZNDFA-SA (IDFA9) = ,70'
AND ZNDFA-VA (IDFA9) = '15'
AND ZNDFA-ST (IDFA9) = '15'
AND ZNDFA-WERT-ZT (IDFA9) NUMERIC
AND ZNDFA-WERT-ZT (IDFA9) = ZERO
AND ZNDFA-SICHER (IDFA9) NUMERIC
AND ZNDFA-SICHER (IDFA9) = ZERO
OR ZNDFA-S-STELLE (IDFA9) = ,RTG'
AND ZNDFA-SA (IDFA9) = ,70'
AND ZNDFA-SICHER (IDFA9) NUMERIC
AND ZNDFA-SICHER (IDFA9) = ZERO
AND (ZNDFA-VA (IDFA9) = ,25'
OR ZNDFA-VA (IDFA9) = '25'
OR ZNDFA-VA (IDFA9) = ,35'
OR ZNDFA-VA (IDFA9) = ,36'
```

Bei 24 Zeilen Umblättern notwendig

```
OR ZNDFA-VA (IDFA9) = ,45'
OR ZNDFA-VA (IDFA9) = ,46')
OR (ZNDFA-S-STELLE (IDFA9) = ,JUZ'
OR ZNDFA-S-STELLE (IDFA9) = ,BWE'
OR ZNDFA-S-STELLE (IDFA9) = ,KLP')
AND ZNDFA-SA (IDFA9) = ,90'
AND ZNDFA-WERT-ZT (IDFA9) NUMERIC
AND ZNDFA-WERT-ZT (IDFA9) = ZERO
AND ZNDFA-SICHER (IDFA9) NUMERIC
AND ZNDFA-SICHER (IDFA9) = ZEP
AND (ZNDFA-VA (IDFA9) = ,25'
OR ZNDFA-VA (IDFA9) = ,30'
OR ZNDFA-VA (IDFA9) = ,32'
OR ZNDFA-VA (IDFA9) = ,33'
OR ZNDFA-VA (IDFA9) = ,34'
OR ZNDFA-VA (IDFA9) = ,35'
OR ZNDFA-VA (IDFA9) = ,36'
OR ZNDFA-VA (IDFA9) = ,45'
OR ZNDFA-VA (IDFA9) = ,46')
    MOVE ,O' TO ZNDFA-OK (IDFA9)
END-IF.
```



Unter welcher **fachlichen** Beding-ung wird  
ZNDFA-OK mit ,O' belegt?

**+50% lfd. Kosten in der AE**

```
FOR Z = 1 TO ANZ // für alle Elemente
  CUR = FIRST;
  WHILE (CUR != NULL && NOT LAST)
    IF CUR->NEXT != NULL
      THEN CUR = CUR->NEXT;
    ELSE
      LAST = '1';
    END;
  ... // APPEND ELEMENT
NEXT Z
```



Insert in  $O(n)$  statt  $O(1)$  ( $\rightarrow$  Listenaufbau:  $O(n^2)$ )  
> 100.000 EUR p.a. Mehrkosten im Rechenzentrum!

# Weitere typische Mängel

- Workarounds
- Inperformante Algorithmen
- Inadäquates Datenmodell
- Abnehmende Datenqualität
- Fehlende Dokumentation
- Fehlendes Know-How im Team
- Infrastruktur vom Hersteller nicht mehr unterstützt
- Werkzeuge entsprechen nicht mehr Stand der Technik
- Fehlende Funktionalität – Anforderungstau



# ... auch in „modernen“ Systemen



```
// If the dialog was closed with „cancel“ the selection
// disappears. I don't know why.
// solution: re-initialize the selection
if (!allowed) {
    final int level = this.getLevel();
    // since the reinit is a change for the controller we
    // detach the listener to avoid recursion
```

In Java-Systemen bereits beobachtet:

- Veraltet Frameworks (Toplink u.a. Persistenzansätze, Struts, ...) (noch bis vor kurzem kein „responsive Design“)
- Überbordene/unsinnige Schichtenarchitekturen
- Die gleiche Bibliothek in mehreren verschiedenen Versionen in ein und derselben Anwendung?
- Wiederholt: Fehler in der Synchronisation von Threads
- Exception Handling ~ ein Zufallskonzept?
- Falsche Anwendung von Design-Patterns
- Hard-coded Pfade, User-Ids u.a.; 2 x 50% Code-Duplikation, 1 x 85% Code-Duplikation



# Wachstum + „Alterung“ ⇒ Kosten

## Negative Einflüsse auf Produktivität:

(C. Jones, 1999)

### Neuentwicklung:

#1 Reuse of poor-quality deliverables **-300%**

### Weiterentwicklung/Wartung:

#1 Error prone modules **-50%**

#2 Embedded variables and data **-45%**

## ⇒ **Exorbitante Mehrkosten und Zeitbedarf**

(auf lange Sicht nicht 10% sondern 500% höhere laufende Kosten!)

## ⇒ **Risiken im globalen Wettbewerb**

(wirken bisher noch wenig, da Schwächen verbreitet - unreife Branche)

## ⇒ **Kein Erfolgsmodell für die Zukunft!**



# ALTERNATIVEN

Was tun?

# Ignorieren, „Einfrieren“, Abwarten

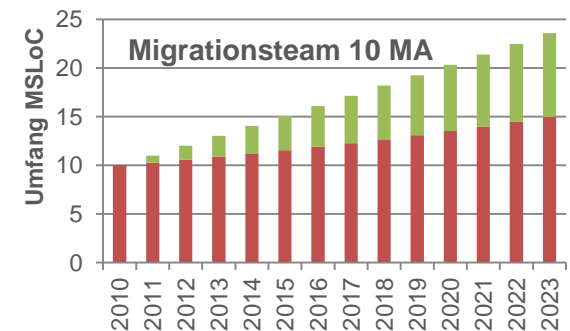
- Alt-Lösung einfrieren und später abschalten
- Beispiel: „alles Neue nur noch in Java“
- Realität:
  - Starkes Wachstum neuer Umfänge (15 -50% p.a.)
  - **Umfänge in Alt-Technologie wachsen weiter ...**  
(Gartner: 5 billion new lines of COBOL every year.)



## Strategie scheitert aufgrund Wachstum „Alt-“Umfänge

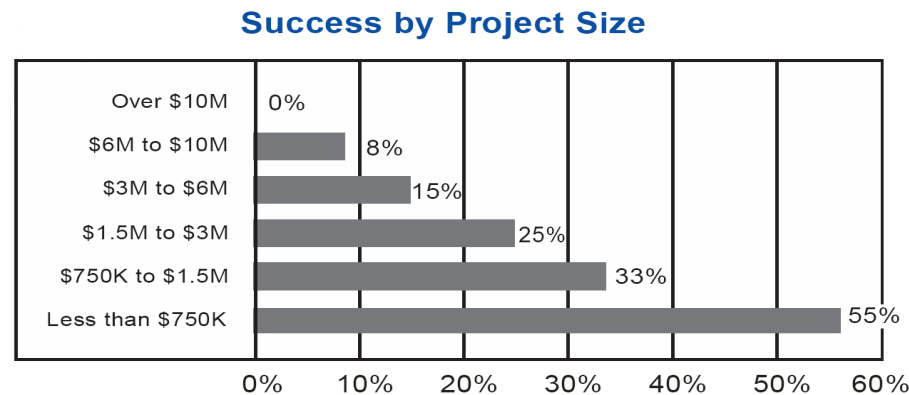
### Analog:

- „Sanfte Migration“ ~ „Abwarten“, falls Geschwindigkeit geringer als Wachstum bzw. Alterung
- „Application Retirement“ ~ „nichts tun“, da nur Anwendungen mit geringer Nutzung abschaltbar sind



## Big Bang Ablösung - „Tabula Rasa“

- Enorme Gefahr a) Unterschätzung der Aufgabe  
b) Entwicklung auf ein „Moving-Target“
- Teuer, zeitaufwendig und riskant



Ablösung ist Vernichtung von **Kapital** – alle 10 Jahre alles neu ist unmöglich

- „Wrapping“, d.h. Schalen um die vorhandenen Komponenten
  - Service-Fassaden
  - Beibehaltung der vorhandenen Geschäftslogik für neue Dienste
  - u.a.

- Kauf eines Standardprodukts

- D.h. „alles neu“ und
- Das Problem soll in Zukunft ein anderer haben und lösen
- ... mit weitreichender Aufgabe von Steuerungsmöglichkeiten



## Wiederherstellung von **Qualität und Effizienz!**

- **zeitgemäße** Qualität (heute bestmöglich)
- ggf. auf **allen Ebenen** (nicht nur Technologie!)
- gemessen an **Wirtschaftlichkeit**  
Nicht: „neueste“ Mode oder subjektive „Schönheit“

## **Modernisiert = ökonomisch zukunftsfähig!**

- **Extrem anspruchsvoll**, bislang kaum Standard-Prozesse/-Antworten
- **Kontinuierliche** Aufgabe

# VORGEHEN

Spezielle Aspekte

- Software Reengineering Assessment Handbook
- US Verteidigungsministerium, 1997
- Betrachtung aller Alternativen für Software
  - Beibehaltung as-is
  - Ersatzlose Abschaltung
  - Fachliche Erweiterung
  - Redokumentation
  - Plattformmigration, ggf. auch durch Übersetzung
  - Partielle Optimierung (z.B. Performance-Optimierung)
  - Vollständiger, technischer Rewrite (jedoch keine Neuentwicklung)
  - ...

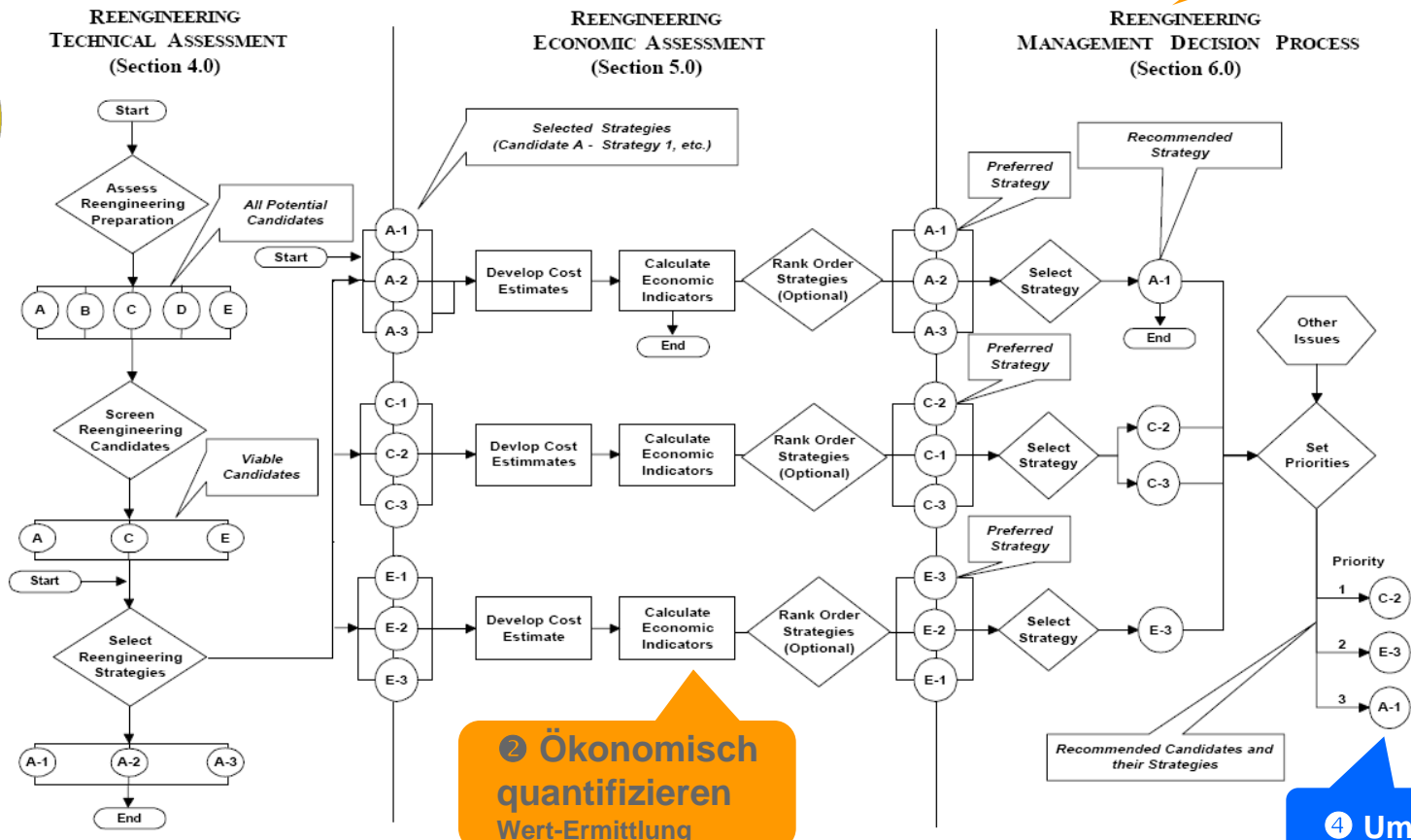


# SRAH Bewertungsprozess

**1 Chancen identifizieren**  
Produkt-/Prozessbewertung

**3 Strategisch absichern**

## SOFTWARE REENGINEERING ASSESSMENT PROCESS OVERVIEW



**2 Ökonomisch quantifizieren**  
Wert-Ermittlung

**4 Umsetzen**



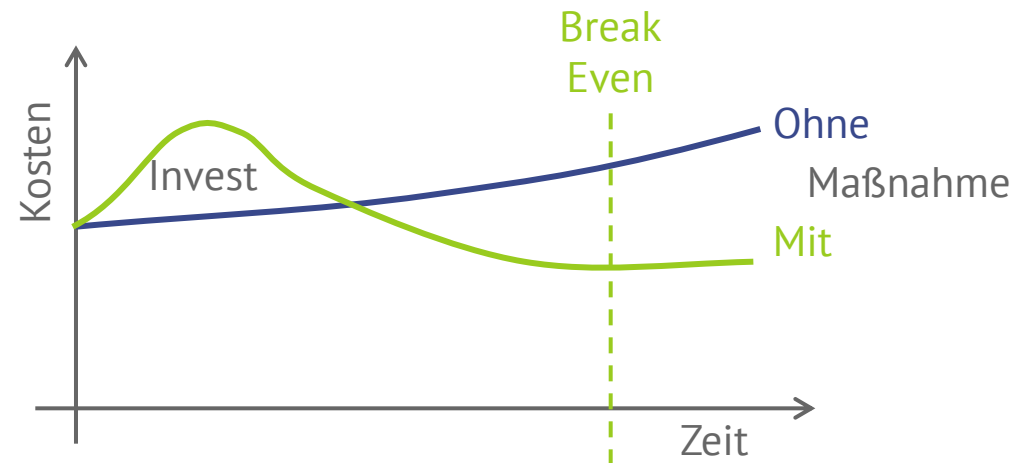
1. Fachlichen und technischen Handlungsbedarf klären
  - Analyse Ist-Zustand
    - Werkzeuge
    - Manuell und Expertise
  - **Zielbild entwickeln (Ideen und Alternativen gefragt!)**
    - Zielarchitektur und Inhalte
    - Qualitätsziele
2. Rahmenbedingungen klären ~ **Budget, Zeit, ROI, ...**
3. **Entwurf Modernisierungsstrategie**
  - **Schnittlinien (fachlich und/oder technisch)**
  - Techniken für die Anbindung alt/neu
  - Aktivitäten
4. Validieren und ggf. justieren ~ Proof of Concept
5. **Iterative Umsetzung** und Einführung mit Mischbetrieb alt/neu
6. Management, Nutzer, Fachbereich früh einbinden ~ PLK, Pilotbetrieb, ...



## A) Festlegung Ist, Soll und Handlungsbedarf

## B) Festlegung und Anordnung der Aktivitäten auf Basis:

1. Ist-Kosten
2. Potenzial
3. Max. Invest-Höhe
4. Zeitpunkt Break-Even
5. Technische Randbedingungen



- ⇒ **Es gibt immer einen realistischen Weg!**
- ⇒ **Sanierung i.d.R. langjährig, zahlreiche Einzelschritte**
- ⇒ **Kein hohes Risiko erforderlich!**

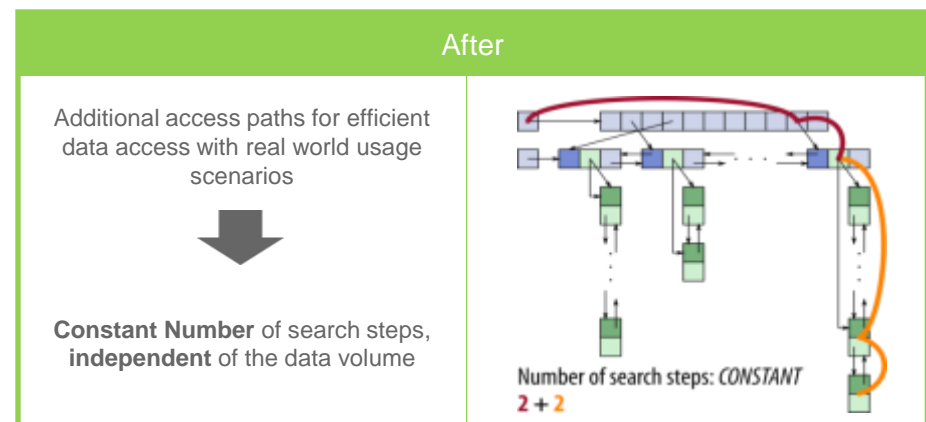
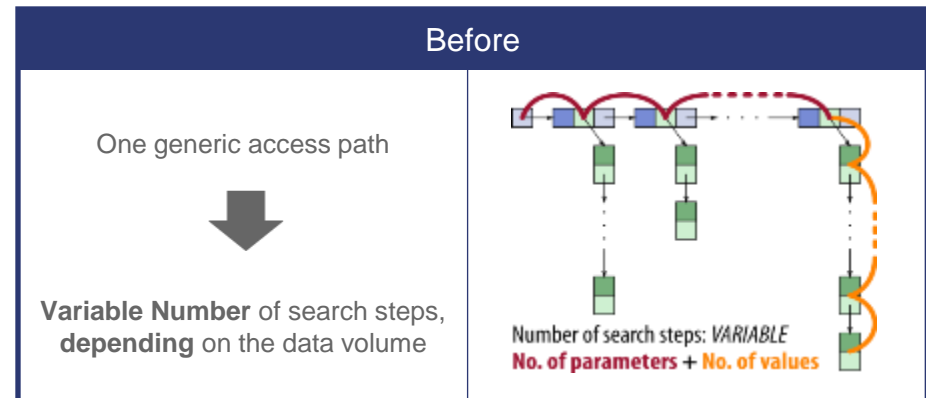
# Achtung: es gibt besondere Anforderungen

- Team
  - „Alte“ Sprachen, Technologien, Idee verstehen
  - Neues Design und Technologien beherrschen und anwenden
- Reverse Engineering
  - Bereitschaft altes zu verstehen
  - Analytische Fähigkeiten
- Test – Vergleich vorher-nachher
- Präzision und Null-Fehler-Mentalität  
(bei vorhandenen Systemen mit Daten sind Fehler besonders schädlich)

# BEISPIELE

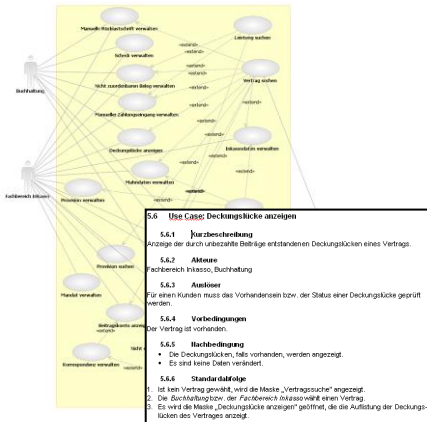
# Beispiel 1: Performance-Optimierung

- Zentrale Banken-IT
  - 129,6 mio Konten
  - 84.100 mio Transaktionen p.a.
  - 10.431 Windows / UNIX Server
  - 20.761 TB Storage
  - **384.039 MIPS z/OS**
- Zentrale Komponente
  - Message broker Portal/Backend
  - Parameter Marshalling
- Beobachtung
  - Werte in single linked lists
- Optimierung
  - Main memory hashing
  - Multiple linked lists und Index
- Ca. 60% CPU Einsparung (> 1 Mio € Kosten p.a.) bei dieser Komponente



# Beispiel 2: Runderneuerung

## Manuelles Reverse Engineering und Rewrite.



Use Cases

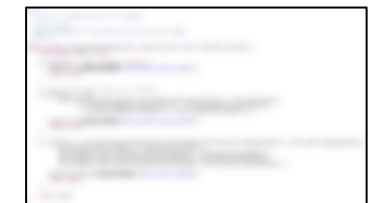
Geschäftsregeln

Id	Abkürzung	Regel
1	Andere Services mit dem Vertrag	...
2	Andere Services mit dem Vertrag	...
3	Andere Services mit dem Vertrag	...
4	Andere Services mit dem Vertrag	...
5	Andere Services mit dem Vertrag	...
6	Andere Services mit dem Vertrag	...
7	Andere Services mit dem Vertrag	...



Dialoge

Informationsmodell



Java-Implementierung

Vollverantwortung, Werk/Festpreis, i.W. ohne Fachbereich

## Altsystem (RPG, AS400)

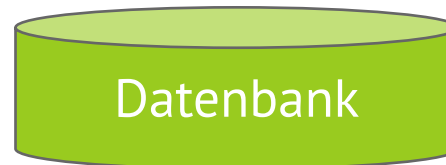


MOVEL	DOCART	TO1DID
MOVE	DOCLFD	TO1DID
MOVEL	#PFAD	TO1FID



Paralleler Zugriff auf gemeinsame Datenbasis

1. Schrittweise Migration Use Cases
2. ca. ¼-jährliche Produktivsetzungen
3. Koexistenz alt/neu
4. Gemeinsame Datenbasis
5. Database-last Migration (DB-Bereinigung als letzter Schritt)



## Zielsystem (Java/Web)



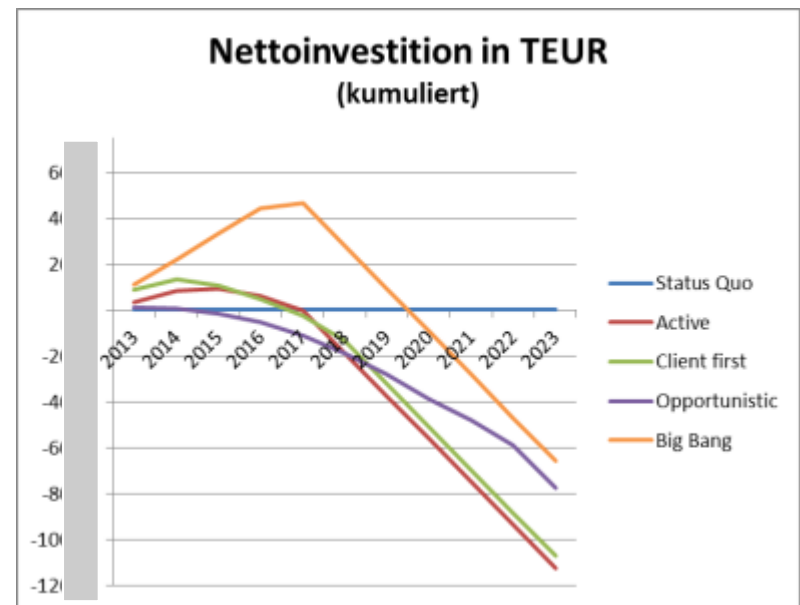
```
// 2. Close current Mandat  
currentlySavedMandat =  
    currentlySavedMandat == null ? mandat :  
    currentlySavedMandat.setDatumUenderungsbeginn(  
    currentlySavedMandat.setMandatsStatus (GESCHLOS  
    currentlySavedMandat.setMandatsUenderungsgrund  
    getValidator().setErrorHandler(errorHandler);  
    getValidator().validate(currentlySavedMandat,  
    if (hasErrorHandlerErrorsOrNotIgnoredWarnings
```



Paralleler Zugriff auf gemeinsame Datenbasis

# Beispiel 3 – Planung einer Roadmap

- Maßnahmen 1: Quick-Win Bereinigung
  - Reduktion des COBOL Code Umfangs um 40%
  - Vollständig neue Build- und Entwicklungsumgebung
  - Überführung fachlicher Abläufe nach Java und Anbindung an Alt-System
- Maßnahmen 2: Redesign Backend COBOL → Java, Services
  - Analyse verschiedener Szenarien
  - Vollständiges **Redesign und Rewrite fachlicher Abläufe** (Berechnungen) auf Basis Ist-Datenbank
  - **Design, Prototyping, Testautomatisierung** von Berechnungsmodellen mittels MS-Excel







HABEN SIE FRAGEN?