

- 2 Durchführung von Reengineering-Projekten
 - Arten von Reengineering-Projekten
 - Initiierung von RE-Projekten im Großen
 - Projektrecthtfertigung
 - Messung von Wartbarkeit
 - Portfolio-Analyse
 - Kostenschätzung
 - Kosten-Nutzen-Analyse
 - Risikofaktoren
 - Häufige Fehler

- Planung und Durchführung von Reengineering-Projekten
- unter Berücksichtigung besonderer Probleme, Risiken und häufiger Fehler

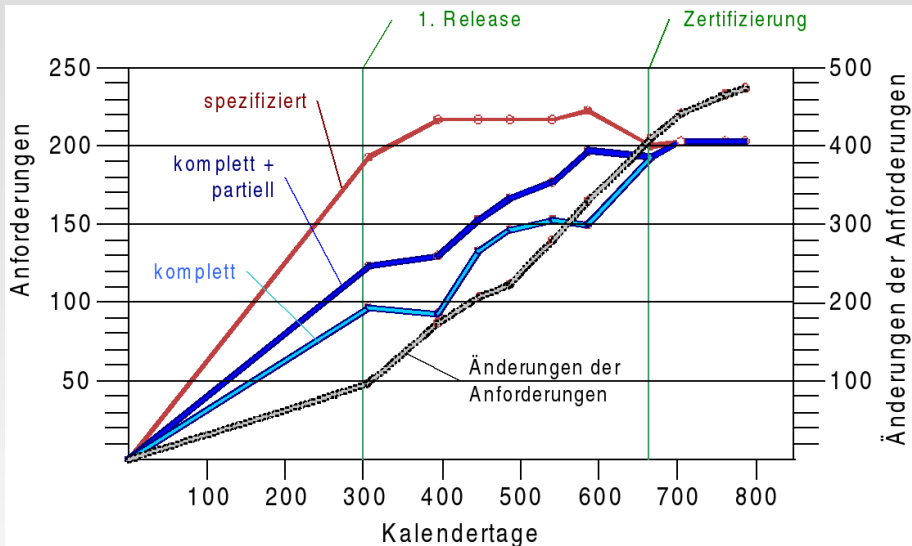
Kontext

- Planung steht ganz am Anfang jedes Projekts
- bisher wurden nur technische Aspekte behandelt

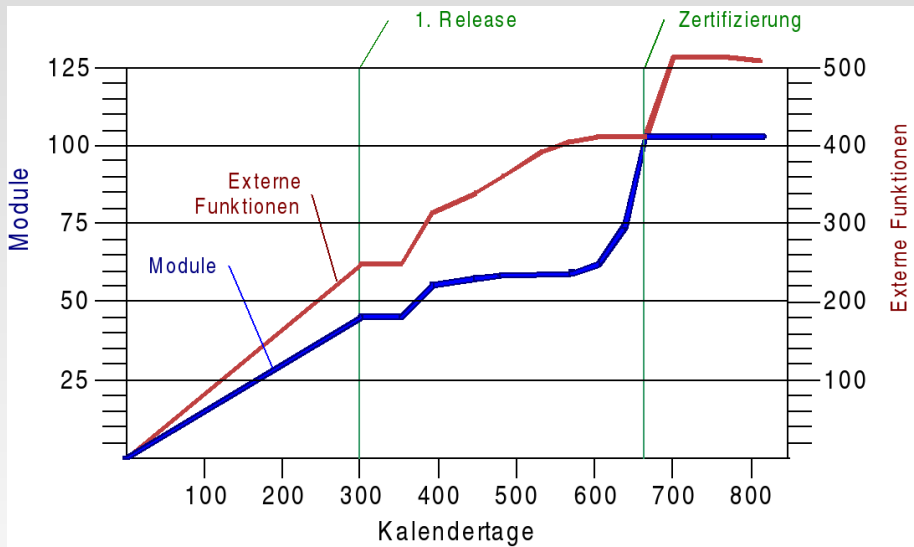
Arten von Reengineering-Projekten

- Reengineering als Teil des normalen Wartungsprozesses:
„Reengineering im Kleinen“
- im Change-Management integriertes Reengineering
- massive Änderung des Codes: „Reengineering im Großen“

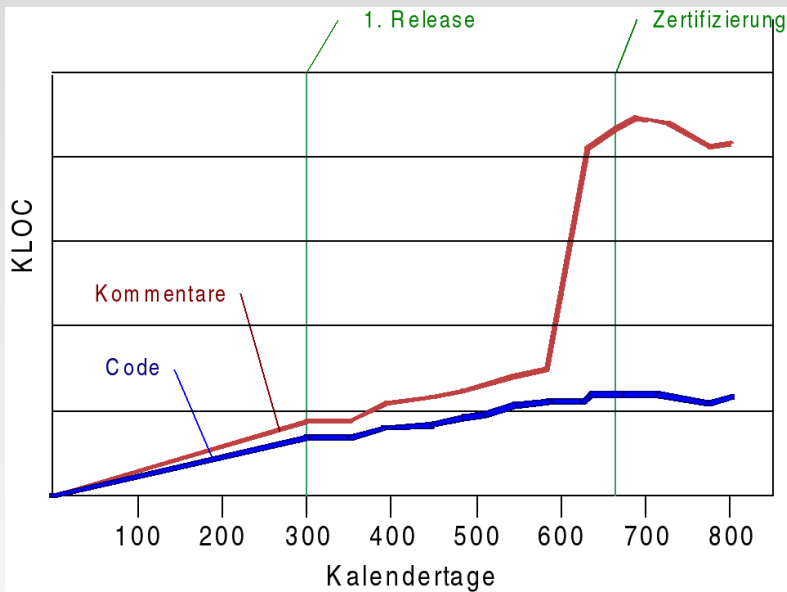
Änderungsrate



Modularisierung



Kommentierung



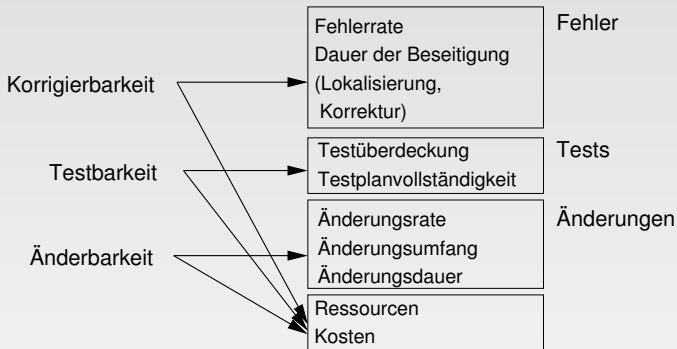
Initiierung von RE-Projekten im Großen nach Sneed (1995)

- ① Projektrechtfertigung
- ② Inventur
 - Welche (Teil-)Systeme existieren (Sprachen, Größe, Einsatz, . . .)?
- ③ Eingrenzung: Portfolio-Analyse
 - Welche (Teil-)Systeme sollen (in welcher Reihenfolge) renoviert werden?
 - Welche Komponenten müssen erweitert werden?
- ④ Kostenschätzung
 - Was wird das Reengineering kosten?
- ⑤ Kosten/Nutzen-Analyse
 - Vergleich von Ausgaben und Einsparungen
- ⑥ Vertragsschluss
 - Definition der Vertragsziele und ihrer Messbarkeit

- Funktionalität wird sich nicht in großem Maße ändern in einem reinen Reengineering-Projekt
 - Wie ist das Projekt zu rechtfertigen?
 - In welchem Maße wird sich die Wartbarkeit verbessern?
 - Bestimme gegenwärtige Qualität
 - Ermittle gegenwärtige Wartungskosten
 - Bestimme Wert der Software (Verkaufserlöse, Wichtigkeit der Kunden, . . .)
 - Schätze zukünftige Qualität, Wartungskosten und Wert ab.
- Zur genauen Bestimmung muss ein Metrikprogramm existieren/etabliert werden:
- Qualität und Wartbarkeit?

Messung von Wartbarkeit (Ideal)

Anhand eines existierenden Prozessmodells für die Wartung werden detaillierte Daten erhoben:



Anhand der Daten wird ein statistisches Modell entwickelt, das zukünftige Wartungsdaten vorhersagt.

Einfache Metriken:

- Anzahl der von Änderung betroffenen Anweisungen
 - z.B. mit 'diff' ermittelt
 - je größer der Wert (relativ zur Größe des Änderungsproblems), desto schlechter ist die Wartbarkeit
- Aufwand
 - Kosten für Personal (Ausrüstung, Testumgebung)
- Anzahl der Fehler hervorgerufen durch Änderungen
 - gefunden durch Regressionstest
 - nach Auslieferung entdeckt
- Traditionelle Software-Qualitätsmetriken
 - McCabe, Complexity, Halstead, ...

Wo beginnen? Portfolio-Analyse

Geschäftswert = Wert der Software

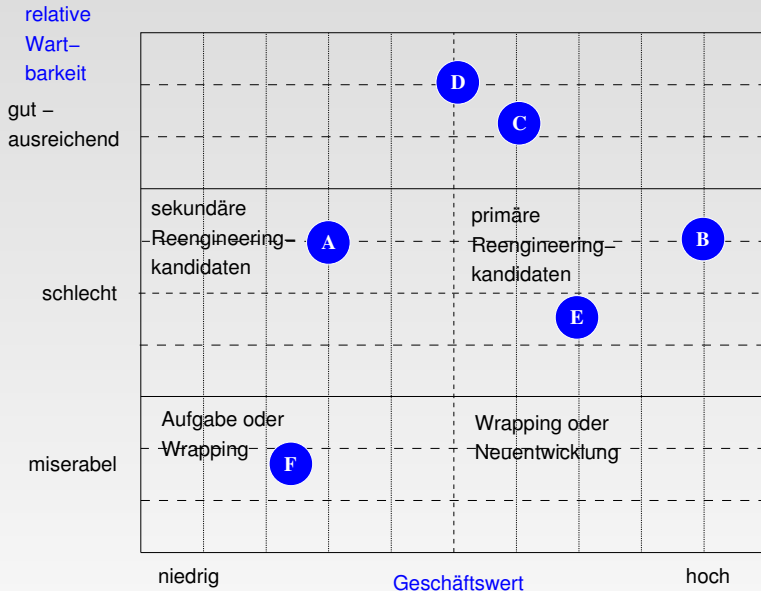
- Bestimmung des Geschäftswerts durch Marktanalytiker
 - Marktwert
 - Beitrag zum Profit
 - Werbewirksamkeit
 - ...
- und Chef-Architekt
 - hängen andere wichtige Systeme davon ab?

Portfolio-Analyse

Vergebe 100 Prozentpunkte pro Wertaspekt. Summiere (eventuell mit Gewichtung) die Zeilen für jedes System.

System	Markwert	Profit	Werbewert	Bewertung
A	10	10	10	30
B	40	30	20	90
C	10	30	20	60
D	10	10	30	50
E	30	20	20	70
Summe	100	100	100	300

Zu sanierende Systeme



Faktoren

- Produkt, z.B.
 - relative Wartbarkeit (wie oben)
 - Metriken der Größe (LOC, Function Points)
 - Aufwand des Testens bzw. bisherige Testüberdeckung (für Regressionstest)
- Ressourcen, z.B.
 - verfügbare Werkzeugunterstützung
 - inklusive Aufwand für Anpassung bzw. Entwicklung von Werkzeugen
 - Erfahrung des Wartungspersonals
- Prozess, z.B.
 - bisheriger Aufwand für ähnliche Projekte, idealerweise bezogen auf einzelne Aktivitäten

Ideal: Aus den oben genannten Faktoren und den früheren Kosten wird mittels statistischer Analysen ein Kostenmodell entwickelt.

Alternative Kostenschätzung nach Sneed (1995)

- Gesamtaufwand = Reengineering-Aufwand + Testaufwand
- $RE\text{-Aufwand } [pm] = \frac{\text{Komplexität} \times \text{Größe } [loc]}{\text{durchschn. Produktivität } [loc/pm]}$ über alle Komponenten
- $\text{Testaufwand} = \# \text{Testfälle} \times \text{Durchschnittsaufwand-pro-Testfall}$
 - $\# \text{Testfälle}$ kann aus zyklometrischer Zahl im Falle der Zweigüberdeckung abgeleitet werden
 - Gewichtung bezüglich Testbarkeit, Testunterstützung, Testumgebung
- Erfahrungswert: $\text{Testaufwand} = n \times RE\text{-Aufwand}$ für $1 \leq n \leq 3$
 - Annahme: Reengineering wird großteils automatisiert.

Projektdauer nach Sneed (1995)

Minimale Projektdauer T (ähnlich zu COCOMO):

$$T = 2,5 \times Aufwand^{0,19}$$

höhere Parallelität als bei Forward-Engineering-Projekten
(COCOMO-Faktor ist 0,38)

Vergleich des (erwarteten) Nutzens von...

- Reengineering
- Neuentwicklung
- Wrapping
- Untätigkeit

- jährliche Wartungskosten
- jährliche Betriebskosten
- jährlicher Geschäftswert
- geschätzte Lebensdauer
- Dauer von Reengineering, Wrapping bzw. Neuentwicklung
- Risiko von Reengineering, Wrapping bzw. Neuentwicklung

Nutzen



Nutzen bei Beibehaltung des Status Quo =
(Geschäftswert – (Wartungskosten + Betriebskosten)) × Lebensdauer

Nutzen der Neuentwicklung =

$$\begin{aligned} & (\text{Geschäftswert} - (\text{Wartungskosten} + \text{Betriebskosten})) \times \text{Entwicklungszeit} \\ & + (\text{Geschäftswert}' - (\text{Wartungskosten}' + \text{Betriebskosten}')) \times (\text{Lebensdauer}' - \text{Entwicklungszeit}) \\ & - (\text{Entwicklungskosten} \times \text{Risiko}) \end{aligned}$$

Handwritten annotations in blue:

- Wartung in d. Phase d. Neuentw. (pointing to the first term's parentheses)
- Einsatz (pointing to the second term's parentheses)
- Kosten Neuentw. (pointing to the third term)

- ohne Berücksichtigung jährlicher Änderungen von Kosten/Nutzen
- Neuentwicklung kann Lebensdauer verlängern

Analoge Betrachtung für Reengineering und Wrapping.

Weitere zu berücksichtigende Faktoren

- Time-To-Market-Pressure
- Neuentwicklung dauert zu lange
- Benutzerzufriedenheit
- kodierte Expertenwissen
- Moral des Wartungspersonals
- Programmierer für neue Sprachen oder Konzepte schwer zu finden
- langjährige Programmierer sträuben sich gegen Neuerung