

Vorlesung Software-Reengineering

Prof. Dr. Rainer Koschke

Arbeitsgruppe Softwaretechnik
Fachbereich Mathematik und Informatik
Universität Bremen

Wintersemester 2006/07

1 Einführung

- Administrativa
- Lernziele
- Motivation
- Wichtige Begriffe
- Wartung
- Reverse Engineering
- Restrukturierung
- Reengineering
- Wrapping
- Business Process Reengineering
- Ziele und Aufgaben
- Unterschiede zur Vorwärtsentwicklung

- Vorlesung: montags, 10:15 - 11:45 Uhr, Raum MZH 5210
- Vorlesung/Übung (alternierend): donnerstags, 8:15 - 9:45 Uhr, Raum MZH
- Erreichbar: OAS, Telefon 218-9671 koschke@tzi.de
- Sprechstunde nach Vereinbarung
- Video im Netz <http://mlecture.uni-bremen.de>
- Literatur: Folien zur Vorlesung und verwendete Artikel
http://www.informatik.uni-bremen.de/st/lehredetails.php?id=&lehre_id=44
- Reengineering-Bibliographie:
<http://www.iste.uni-stuttgart.de/ps/reengineering/>

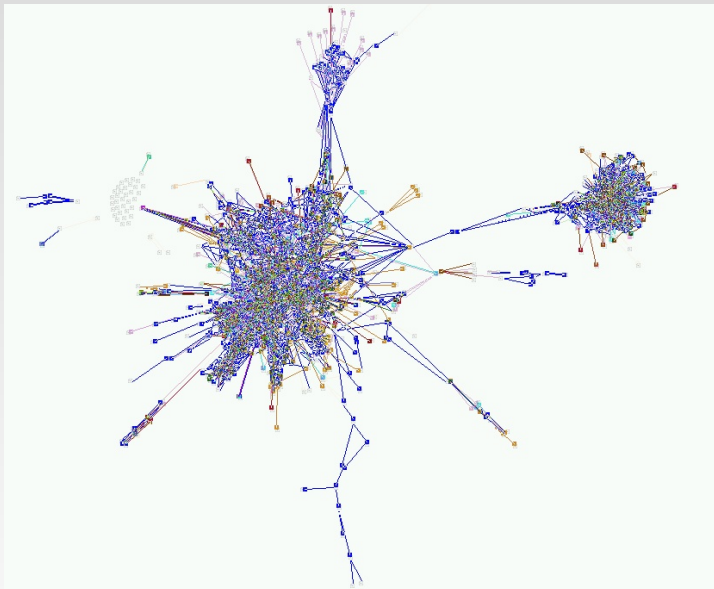
- Modulprüfung: mündliche Prüfung
- ansonsten
 - ① erfolgreiche Bearbeitung von vier praktischen Aufgaben:
 - ① Qualitätsanalyse (quellcodenah)
 - ② Refactoring/Transformation
 - ③ Architekturprüfung
 - ④ Merkmalsuche
 - ② Fachgespräch

- Grundlegende Begriffe
- Übersicht über die Gebiete des Reengineerings
- Abgrenzung zum traditionellen Software-Engineering

Software steuert Systeme



Ein Blick unter die Haube



Software-Evolution

- Gesetz des fortgesetzten Wandels
- Gesetz der ansteigenden Komplexität
- ...

⇒ ständige Anpassung erforderlich

⇒ Komplexität muss kontrolliert und begrenzt werden

- Gewählte Lösung antizipiert mögliche Änderungen.
- Änderungen werden auf der adäquaten Ebene vorgenommen.
- Dokumentation wird mitgeführt.

- Die Zukunft lässt sich nur begrenzt vorhersagen.
- Ursprüngliche Systemstruktur wird ignoriert.
- Dokumentation ist unvollständig oder obsolet.
- Mitarbeiter verlassen das Projekt (und mit ihnen verschwindet das ganze Wissen).

Legacy:

“A sum of money, or a specified article, given to another by will; anything handed down by an ancestor to a predecessor.”

– Oxford English Dictionary

Legacy:

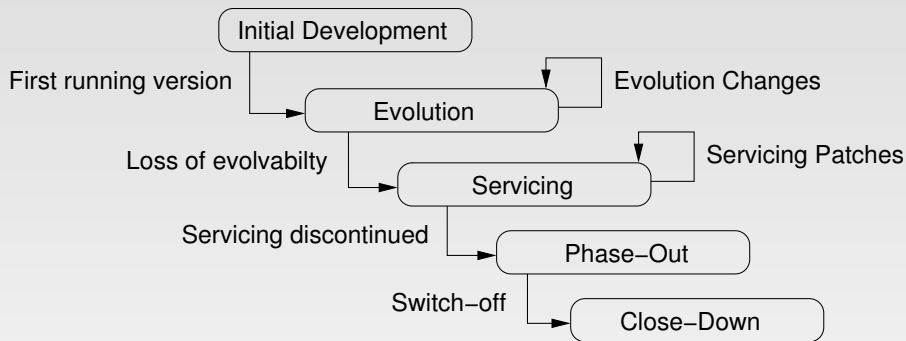
“A sum of money, or a specified article, given to another by will; anything handed down by an ancestor to a predecessor.”

– Oxford English Dictionary

Definition

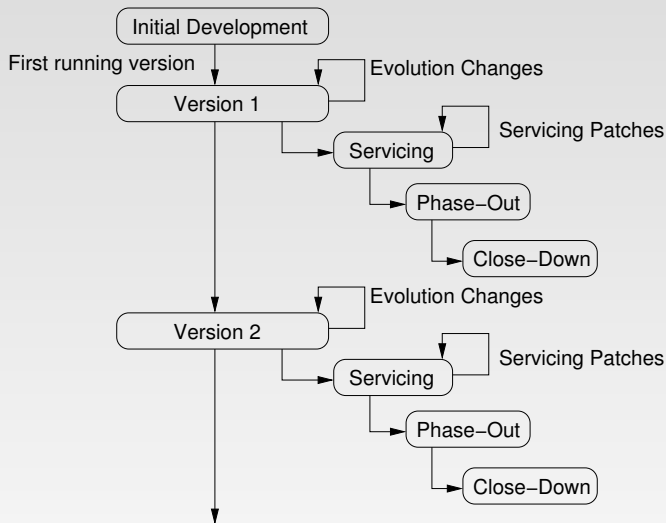
Legacy System: Software-System, das geerbt wurde und einen Wert darstellt.

Staged Software Life Cycle Model



– Rajlich und Bennett (2000)

Versioned Staged Model (Rajlich und Bennett 2000)



- Software-Wartung
- Software-Evolution
- Reengineering
- Software-Reengineering
- Business-Process-Reengineering
- Renovation
- Reclamation
- Reverse Engineering
- Restrukturierung
- Wrapping

ANSI/IEEE Standard 729-1983:

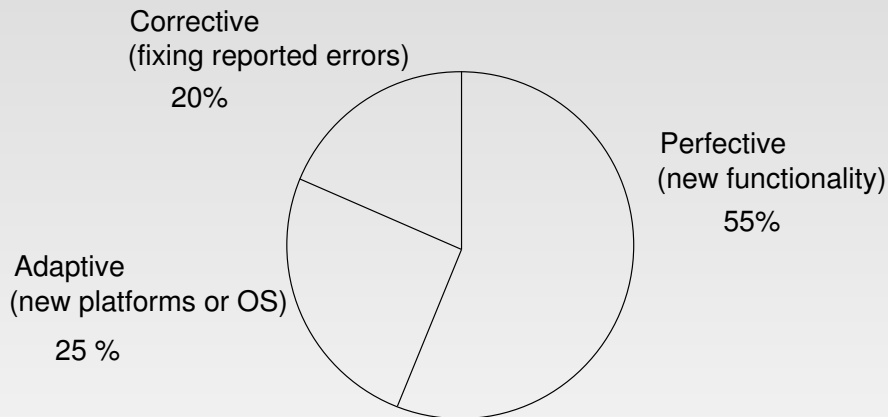
“Modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a changed environment.”

Häufiger Sprachgebrauch: Änderungen am System nach dessen Auslieferung.

Schließt Anpassungen an neue Anforderungen ein.

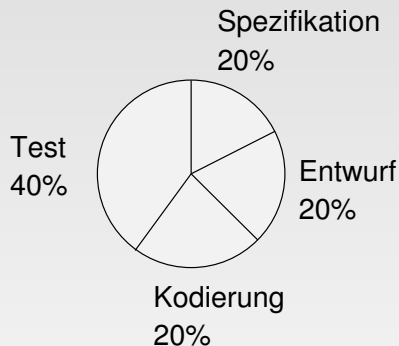
Besserer Begriff hierfür: Software-Evolution.

Aufwand für Software-Wartung



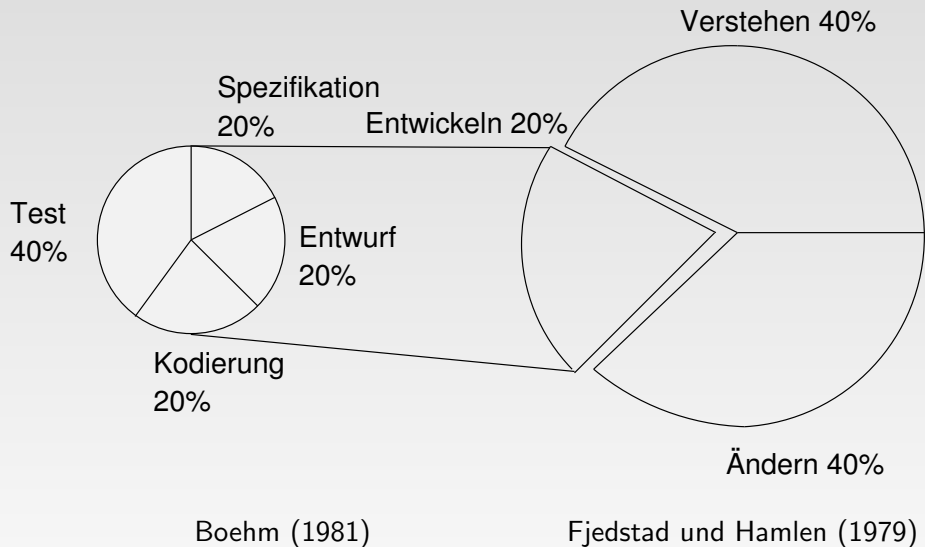
– Lientz und Swanson (1980)

Aufwand im Software-Lifecycle



Boehm (1981)

Aufwand im Software-Lifecycle



US Air Force System (Boehm, 1975):

- \$ 30 / Statement bei Erstentwicklung

US Air Force System (Boehm, 1975):

- \$ 30 / Statement bei Erstentwicklung
- \$ 4000 / Statement in der Wartung