

Allgemeines

<i>Dozent:</i>	Prof. Dr. Gerd Beneken
<i>Verantwortlich:</i>	Prof. Dr. Gerd Beneken
<i>Studiengang:</i>	Master
<i>Pflicht/FWPF:</i>	FWPF
<i>Voraussetzungen:</i>	vertiefte Kenntnisse im Software Engineering
<i>Sprache:</i>	Deutsch
<i>Lehrform:</i>	4 SWS Seminaristischer Unterricht mit Übungen
<i>Arbeitsaufwand:</i>	60 Stunden Präsenzzeit, 90 Stunden Selbststudium
<i>Leistungspunkte:</i>	5
<i>Medienform:</i>	Vortrag mit Overhead-Projektor und Laptop mit Beamer, Vorführung von Programmbeispielen, Einzelbetreuung während der Übungen im Labor, Skriptum in Folienform
<i>Prüfung:</i>	Klausur, 90 Minuten, Unterlagen lt. Aushang

Lernziele und Inhalt

Richtziel

Die Studierenden kennen die wichtigsten Techniken und Verfahren der Wartung- und Weiterentwicklung großer Softwaresysteme. Die Studierenden können bestehende Software analysieren und den Aufwand zu ihrer Erhaltung oder Ablösung beurteilen. Sie können die Wartung und Weiterentwicklung organisieren und auch selbst durchführen. Sie beherrschen die Werkzeuge und Methoden des Debugging, des Refactoring, des Performance Tuning, des Reengineering sowie des Regressionstests.

Inhaltsübersicht

Diese Veranstaltung vertieft die in den Bachelor-Fächern Software Engineering 1 und 2 erworbenen Kenntnisse der Software-Wartung und –Weiterentwicklung. Grundlegende Begriffe und Techniken werden dargestellt und an Fallstudien erarbeitet und eingeübt.

Inhalt

- 1. Grundlagen*
 - *Der Software-Lebenszyklus*
 - *Legacy-Systeme, Legacy Frameworks und Bibliotheken*
 - *Kategorien von Wartung (Adaptiv, Korrektiv, Präventiv, Perfektionierend)*
- 2. Evolution von Software*
 - *Alterung*
 - *Wachstum*
 - *Entropie*
- 3. Wartungsprozesse*
 - Ablauforganisation, Rollen, Produkte*
 - Kanban*
 - Konfigurationsmanagement*
 - Qualitätsmanagement*
 - Dokumentation*
- 4. Softwareanalyse und -visualisierung*
 - Programmverstehen*

- Software-Visualisierung*
Software- und Architekturanalyse
5. *Reengineering, Sanierung und Migration*
Reverse-Engineering, Forward-Engineering, Restrukturierung
Renovierung, Sanierung und Refactoring
Migration
 6. *Test und QS in der Wartung*

Literatur

Besonders empfohlen

1. Bommer, Spindler, Barr.: *Softwarewartung*. dpunkt-Verlag (2008)
2. Sneed, Hasitschka, Teichmann: *Software-Produktmanagement: Wartung und Weiterentwicklung bestehender Anwendungssysteme*, dpunkt-Verlag (2004)
3. Feathers: *Working Effectively with Legacy Code*, Prentice-Hall, (2004)

Zusätzlich empfohlen

4. Popp: *Konfigurationsmanagement*, dpunkt-Verlag (2007)
5. Masak: *Legacysoftware: Das lange Leben der Altsysteme*, Springer (2007)