

Allgemeines

<i>Dozent:</i>	Prof. Dr. Gerd Beneken, Prof. Dr. Martin Deubler
<i>Verantwortlich:</i>	Prof. Dr. Gerd Beneken
<i>Studiengang:</i>	Bachelor
<i>Pflicht/FWPF:</i>	Pflicht
<i>Voraussetzungen:</i>	Programmieren 3, Software-Engineering 1
<i>Sprache:</i>	Deutsch
<i>Lehrform:</i>	2 SWS seminaristischer Unterricht und 4 SWS Projektarbeit
<i>Arbeitsaufwand:</i>	90 Stunden Präsenzzeit, 120 Stunden Selbststudium
<i>Leistungspunkte:</i>	7
<i>Medienform:</i>	Vortrag mit Overhead-Projektor und Laptop mit Beamer, Projektarbeit im Team mit Präsentation der Ergebnisse, Projektbetreuung während der Übungen im Labor, Skript
<i>Prüfung:</i>	Studienarbeit mit mehreren Vorträgen

Lernziele und Inhalt

Richtziel

Die Studierenden können selbstständig in kleinen Team Software-Entwicklungsprojekte durchführen. Sie sind in der Lage, in größeren Projekten die Rollen Entwickler, Tester, Qualitätsverantwortlicher, Projektleiter oder Architekt auszufüllen. Die Studierenden beherrschen die dazu notwendigen fachlichen und technischen Grundlagen. Sie können die grundlegenden Methoden des Requirements-Engineering, der Qualitätssicherung und der Software-Architektur anwenden. Sie können ihre Entwürfe mit einer Programmiersprache wie Java, C# oder PHP praktisch in einem Entwicklungsteam umsetzen und beherrschen die dazu notwendigen Werkzeuge, etwa zum Issue Tracking oder zum Konfigurationsmanagement. Zusätzlich trainieren die Studierenden wichtige Soft-Skills, etwa Teamfähigkeit in selbstorganisierten Teams, Präsentationen halten, Feedback geben und empfangen, Umgang mit Konflikten sowie Reviews und Meetings organisieren und moderieren.

Inhaltsübersicht

Die Studierenden führen realitätsnahe Softwareentwicklungsprojekte auf der Basis objektorientierter Programmierung und der in *Software-Engineering 1* und *Programmieren 1-3* erworbenen Kenntnisse durch. Beispiele für Projekte sind: Steuerungs-Software für die Lichtwerkstatt der Rosenheimer Innenarchitekten, Stundenplanungs-Software für die Hochschule Rosenheim oder Erweiterungen für die Rosenheimer Community-Plattform.

Die Projekte werden von den Studierenden in Teams mit drei bis vier Teilnehmern durchgeführt. Ziel ist jeweils eine vollständige Anwendung zu erstellen von der Anforderungsanalyse bis zum Abnahmetest. Als Auftraggeber fungieren externe Partner aus der Forschung oder der Wirtschaft sowie Mitarbeiter der Hochschule.

Jedes Team übernimmt zwei Rollen:

- Die Rolle der Qualitätssicherung (Mitwirkung bei Anforderungsanalyse; verantwortliche Durchführung der Abnahme),
- Die Rolle des Auftragnehmers (Durchführung des Projekts bis zur Abnahme).

Jedes Team erstellt einen Projektbericht und präsentiert den Projektfortschritt in mehreren Kurzpräsentationen, zusätzlich werden mehrere Projektmeetings sowie Reviews der entstehenden Dokumente und des Codes durchgeführt. Die Studierenden in der Rolle des Auftragnehmers präsentieren das laufende System. Der Dozent kontrolliert und steuert die Projekte als Qualitätsmanager.

Inhalt

1. Arbeitstechniken in Software-Entwicklungsprojekten
 - Konfigurationsmanagement mit Subversion
 - Teamorganisation unterstützt durch Excel, Issue Tracker und andere Werkzeuge
2. Kurze Wiederholung: Requirements-Engineering und Spezifikation (Software-Eng. 1)
3. Software-Qualitätssicherung
 - Konstruktive und analytische QS
 - Durchführung von Reviews
4. Software-Architektur
 - Architekturtreiber
 - Prinzipien des Entwurfs
 - Komponenten und Schnittstellen, A-Architektur
 - Umsetzung von Qualitätsanforderungen, T-Architektur
 - UML 2.x, Architektursichten, Dokumentation von Architekturen
5. Implementierung
 - Software-Entwicklungsumgebungen
 - Test-Driven Development und Modultest
6. Software-Qualitätssicherung II
 - Software-Test
 - Code-Reviews und statische Analyse
7. Integration und Abnahme
 - Akzeptanzkriterien, Abnahmetest
8. Wartung und IT-Betrieb
9. Juristische Aspekte des Software-Engineering
 - Urheberrecht, Patentrecht
 - Haftungsrecht
 - IT-Verträge

Literatur

Besonders empfohlen

1. Siedersleben, J.: Moderne Softwarearchitektur. Dpunkt, 2004
2. Balzert: Lehrbuch der Softwaretechnik: Softwaremanagement, 2. Auflage, Spektrum 2008
3. Siedersleben, J. (Hrsg.): Softwaretechnik. Hanser, 2002

Zusätzlich empfohlen

4. Vigneshow, Schneider, Meyrose: Soft Skills für Softwareentwickler: Fragetechniken, Konfliktmanagement, Kommunikationstypen und -modelle, 2. Auflage, dpunkt, 2010
5. Balzert: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements-Engineering, 3. Auflage, Spektrum 2009
6. Brooks, F.: Vom Mythos des Mann-Monats, MITP, 2003
7. DeMarco, T.: Der Termin, Hanser, 2005
8. Hunt, A. und Thomas, D.: Der pragmatische Programmierer, Hanser, 2003