

Allgemeines

<i>Dozent:</i>	Prof. Dr. Gerd Beneken
<i>Verantwortlich:</i>	Prof. Dr. Gerd Beneken
<i>Studiengang:</i>	Bachelor
<i>Pflicht/FWPF:</i>	FWPF
<i>Voraussetzungen:</i>	Schwerpunktfächer Software-Engineering, Programmieren 3 Software-Engineering 1 und 2
<i>Sprache:</i>	Deutsch
<i>Lehrform:</i>	6 SWS seminaristischer Unterricht und Projektarbeit
<i>Arbeitsaufwand:</i>	90 Stunden Präsenzzeit, 150 Stunden Selbststudium
<i>Leistungspunkte:</i>	8
<i>Medienform:</i>	Projektarbeit im Team unter Anleitung des Dozenten, Referate der Teilnehmer, Vorträge zu ausgewählten Themen, Projektbetreuung während der Übungen im Labor,
<i>Prüfung:</i>	Studienarbeit mit Präsentationen

Lernziele und Inhalt

Richtziel

Die Studierenden erwerben die Fähigkeit zur selbstständigen Lösung von komplexen, praxisrelevanten Problemstellungen im Team für externe Auftraggeber. Die Studierenden wiederholen und vertiefen dabei die wichtigsten Methoden des Software-Engineering. Der Unterschied zwischen dokumentgetriebenen Vorgehensmodellen (Vorgehensmodell in SE2) und agilen Vorgehensmodellen wie Scrum (Vorgehensmodell in DAS) wird deutlich. Sie üben zusätzlich praxisrelevante Softskills wie Teamarbeit, Feedback geben und empfangen sowie Präsentationen halten. Ebenso erlernen Sie den Umgang mit einem externen Auftraggeber.

Inhaltsübersicht

Die Teilnehmer führen in Gruppen von drei bis fünf Mitgliedern kleine, aber wissenschaftlich oder technisch anspruchsvolle Projekte von der Anforderungsanalyse bis zur Inbetriebnahme in eigener Verantwortung durch. Einige Beispiele für Projekte sind:

- Online-Shop für Fahrzeugteile mit Java EE
- Mitgliederverwaltung für Kreisjugendring Berchtesgaden in .NET 3.5
- Prozessunterstützung für Entwicklungsprojekte mit Microsoft Sharepoint
- Repository-Mining auf JIRA- und Subversion-Repositories
- Continuum-Integration Plattform für Rosenheimer Teamserver
- Projekt-Controlling Software für ein kleines Unternehmen mit Microsoft DSL-Tools

In allen Fällen sind Dokumentation und Präsentation der Zwischenergebnisse wichtiger Teil des Projekts. Die Veranstaltung ist abgestimmt auf die Veranstaltungen Software-Engineering 1 und 2 sowie Programmieren 1 bis 3.

Die Projekte werden in drei Iterationen (Sprints) durchgeführt, beginnend mit einem explorativen Sprint, in dem ein technischer Durchstich entsteht. Darauf folgen zwei Sprints in denen User-Stories umgesetzt werden.

Inhalt

1. Einführung in Scrum
 - Ziele agiler Vorgehensmodelle
 - Rahmenprozess Scrum
 - Rollen: Team, Product Owner, Scrum Master
2. Anforderungen in Scrum
 - User-Stories und Akzeptanztests
 - Product Backlog
 - Sprint Backlog
3. Planung, Durchführung, Abschluss
 - Sprint-Planung, Planning Poker
 - Tasks erstellen, Taskboard pflegen
 - Daily Scrum
 - Sprint Review-Meeting
 - Sprint Retrospektive

Die Grundlagen von Scrum werden während der Projektdurchführung vermittelt. Jeder der drei Sprints wird nach den Regeln von Scrum durchgeführt.

Literatur

Besonders empfohlen

1. Wirdemann: Scrum mit User-Stories, Hanser, 2009
2. Siedersleben J.: Moderne Softwarearchitektur. Dpunkt, 2004
3. Seifert, J.: Visualisieren. Präsentieren. Moderieren, Gabal, 2005

Zusätzlich empfohlen

4. Cohn: Agile Estimating and Planning, Prentice Hall, 2005
5. Kerth, Project Retrospectives: A Handbook for Team Reviews, Computer Bookshops, 2001
6. Poppendiek: Lean Software Development: An Agile Toolkit for Software Development Managers, Addison-Wesley, 2005
7. Boehm, Turner: Balancing Agility and Discipline, Addison-Wesley, 2004