

Software Engineering

Lernergebnisse

Die Teilnehmer lernen in der Vorlesung die notwendigen Techniken, Tools und Frameworks kennen, die zur Erstellung und zum Management von großen und komplexen Software-Produkten notwendig sind. In den Übungen wird das Erlernte zum Entwerfen einer modular aufgebauten Datenbankanwendung, bei der mehrere "Softwareentwickler" involviert sind, angewendet. Die Teilnehmer koordinieren und verwalten verschiedene Ressourcen und gestalten die zu entwickelnde Softwareanwendung mit Hilfe diverser Frameworks.

Inhalte

- Ziele des Software-Engineering
- XML und XML-Technologien (XML-Schema, XPath, XSLT)
- JSON
- UML (Anwendungsfall Diagramme, Aktivitätendiagramme, Sequenzdiagramme)
- Phasen des Softwareentwicklungsprozesses
- Konfigurationsmanagement
- agile Prinzipien
- agile Methoden (Paarprogrammierung, testgetriebene Softwareentwicklung, XP)
- agile Vorgehensmodelle (Feature Driven Development, Scrum)
- Datenbankanbindung u. Objekt-Relationales Mapping
- Modularisierung
- Prinzipien für objektorientiertes Paketdesign
- Qualitätsaspekte im Softwareentwicklungsprozess
- Klassifikation von Softwaretestmethoden
- (Automatisiertes-) Testen von Programmen
- Continuous Delivery Deployment.

Literatur

Balzert, Helmut (2005): Lehrbuch Grundlagen der Informatik. Konzepte und Notationen in UML 2, Java 5, C++ und C#, Algorithmik und Software-Technik, Anwendungen; mit CD-ROM und e-learning-Online-Kurs. 2. Aufl. Heidelberg, München: Elsevier, Spektrum, Akad. Verl. (Lehrbücher der Informatik).

Balzert, Helmut (2008): Lehrbuch der Softwaretechnik. 2. Aufl. Heidelberg: Spektrum, Akad. Verl.

Balzert, Helmut; Ebert, Christof (2008): Softwaremanagement. 2. Aufl. Heidelberg: Spektrum Akad. Verl. (Lehrbücher der Informatik, / Helmut Balzert ;). Online verfügbar

unter deposit.d-nb.de/cgi-bin/dokserv.

Coad, Peter; Yourdon, Edward (1994): OOD. Objektorientiertes Design. 1. Aufl.

Coad, Peter; Yourdon, Edward (1996): OOA. Objektorientierte Analyse. 2. Aufl. New York, NY: Prentice-Hall.

Rupp, Chris; Queins, Stefan; die SOPHISTen, die (2012): UML 2 glasklar. Praxiswissen für die UML-Modellierung. 1., aktualisierte und erweiterte Auflage, neue Ausg. München: Hanser, Carl.

Lehrmethode

Vorlesung mit gemischten Medien (Folien, Tafelarbeit, Demonstrationen, etc.), Übungen u.a. im Labor, am Computer etc.

Lehrsprache

Deutsch, Englisch

Studien-/Prüfungsleistung

Klausur, Projektarbeit oder sonstige schriftliche Arbeit

Credits

5 (150 h = 60 h Präsenz-und 90h Eigenstudium)

Besonderes

Exemplarische Transferaufgaben:

- Aktive Mitarbeit an der Konzeption oder Planung eines Softwareprojektes bzw. Einführung einer betrieblichen Anwendung
- Aktive Mitarbeit in der Projektkoordination eines Softwareprojektes bzw. bei der Einführung einer betrieblichen Anwendung
- Aktive Mitarbeit in der Programmierung einer Software