

Allgemeines

<i>Dozent:</i>	Dipl.-Inf. (FH) Josef Adersberger
<i>Verantwortlich:</i>	Prof. Dr. Reiner Hüttl
<i>Studiengang:</i>	Master
<i>Pflicht/FWPF:</i>	FWPF
<i>Voraussetzungen:</i>	Gute Programmierkenntnisse insbesondere in Java
<i>Sprache:</i>	Deutsch
<i>Lehrform:</i>	2 SWS Seminaristischer Unterricht mit Übungen
<i>Arbeitsaufwand:</i>	30 Stunden Präsenzzeit, 45 Stunden Selbststudium
<i>Leistungspunkte:</i>	2,5
<i>Medienform:</i>	Vortrag mit Overhead-Projektor und Laptop mit Beamer, Vorführung von Programmbeispielen, Einzelbetreuung während der Übungen im Labor, Skript in Folienform
<i>Prüfung:</i>	Studienprojekt; Klausur, 60 Minuten, Unterlagen lt. Aushang

Lernziele und Inhalt

Richtziel

Die Studierenden erwerben vertiefende Kenntnis der Prinzipien und Techniken von Werkzeugen des Software-Engineering. Sie kennen die wichtigsten Produkte in diesem Bereich und beherrschen die Anwendung von Werkzeugen in exemplarischen Projekten. Im Rahmen eines Projektes trainieren die Studierenden wichtige Soft-Skills, etwa die inhaltliche Strukturierung, Umsetzung und Präsentation eines Projekts.

Inhaltsübersicht

Der Schwerpunkt dieser Veranstaltung liegt auf Konfigurations- und Buildmanagement sowie der automatisierten Qualitätssicherung. Die wichtigsten Verfahren und Werkzeuge für die effiziente Durchführung von Projekten werden beschrieben und eingeübt. Die verwendete Programmiersprache ist Java.

Folgende Themen werden behandelt und in jeweils einem Übungsblock an Beispielen vertieft:

- Einführung und Abgrenzung der Disziplin Software-Entwicklungsumgebung im Bereich des Software-Engineerings.
- Buildmanagement
- Konfigurationsmanagement
- Qualitätssicherung: Dynamische Analysen, Testabdeckung, Profiling
- Qualitätssicherung: Statische Analysen, Metriken
- Generatoren

Inhalt

1. *Grundlagen Software-Entwicklungsumgebungen*
 - 1.1 Einordnung innerhalb des Software-Engineerings
 - 1.2 Strukturierung und Aufbau einer Software-Entwicklungsumgebung
2. *Konfigurationsmanagement*
 - 2.1 Grundlagen des Konfigurationsmanagements: Branching, Tagging, Merging; Umgang mit parallelen Bearbeitungen.
 - 2.2 Vorstellung Versionskontrollsystem: Subversion
 - 2.3 Vorstellung Changemanagementsystem: Trac
3. Buildmanagement
 - 3.1 Anforderungen an das Buildmanagement
 - 3.2 Vorstellung und Vergleich der Werkzeuge Ant und Maven
3. *Qualitätssicherung*
 - 3.1 Grundlagen der Qualitätssicherung
 - 3.2 Dynamische Analysen: Methoden und Werkzeuge des strukturellen Testens
 - 3.3 Statische Analysen: Metriken, Code-Anomalien, Software-Visualisierung
4. *Generatoren*
 - 4.1 Einführung in Generierungstechniken

Literatur

Besonders empfohlen

1. Popp, G.: Konfigurationsmanagement mit Subversion, Ant und Maven. Dpunkt (2006)
2. Simon, F. et al.: Code Quality Management. Dpunkt (2006)
3. Kan, S.H.: Metrics and Models in Software Quality Engineering. Addison-Wesley (2002)

Zusätzlich empfohlen

4. Collins-Sussman, B. et al.: Versionskontrolle mit Subversion (2006)
5. Mason, M.: Pragmatic Version Control Using Subversion. Pragmatic Bookshelf (2006)
6. Tilly, J. und Burke, E.M.: *ANT the definitive Guide*. O'Reilly (2002)
7. Massol V. und O'Brien T.: Maven. A Developer's Notebook (2005)
8. Hunt, A. und Thomas, D.: Pragmatic Unit Testing In Java With JUnit. Pragmatic Bookshelf (2004)