Software-Engineering I (W3WI_SE303)

Software Engineering I

FORMALE ANGABEN ZUM MODUL

MODULNUMMER	VERORTUNG IM STUDIENVERLAUF	MODULDAUER (SEMESTER)	MODULVERANTWORTUNG	SPRACHE
W3WI_SE303	1. Studienjahr	2	Professor Dr. Dietmar Ratz	Deutsch/Englisch

EINGESETZTE LEHRFORMEN

Vorlesung, Übung, Laborübung, Case Study

EINGESETZTE PRÜFUNGSFORMEN

PRÜFUNGSLEISTUNG	PRÜFUNGSUMFANG (IN MINUTEN)	BENOTUNG
Portfolio	Siehe Pruefungsordnung	ja

WORKLOAD UND ECTS-LEISTUNGSPUNKTE

WORKLOAD INSGESAMT (IN H)	DAVON PRÄSENZZEIT (IN H)	DAVON SELBSTSTUDIUM (IN H)	ECTS-LEISTUNGSPUNKTE
150	60	90	5

QUALIFIKATIONSZIELE UND KOMPETENZEN

FACHKOMPETENZ

Die Studierenden kennen gängige Entwicklungsumgebungen und Tools für das Testen und Debuggen von Programmen sowie das Dokumentieren und Konfigurieren von Software. Sie besitzen grundlegende Kenntnisse der Software-Ergonomie, der funktionalen Programmierung und moderner Streaming-Konzepte für Datenstrukturen. Ferner kennen sie zugehörige Entwurfsprozesse und -prinzipien.

METHODENKOMPETENZ

Die Studierenden können die Werkzeuge und Tools sowie die erlernten Konzepte anwenden und autonom kleine bis mittlere Programme mit grafischer Benutzungsoberfläche implementieren, testen und debuggen.

PERSONALE UND SOZIALE KOMPETENZ

Die Studierenden können eigenständig passende Lösungen erarbeiten. Sie können stichhaltig und sachangemessen über Konzepte und eigene Implementierungen und damit verbundene Probleme argumentieren, eigene Umsetzungen plausibel darstellen und eventuelle Fehler nachvollziehbar gegenüber anderen begründen.

ÜBERGREIFENDE HANDLUNGSKOMPETENZ

Die Studierenden können unter Einsatz einer Programmiersprache einfache praktische Aufgabenstellungen bearbeiten, in modernen Technologien umsetzen und damit anwenderfreundliche Lösungen entwickeln.

LERNEINHEITEN UND INHALTE

LEHR- UND LERNEINHEITEN	PRÄSENZZEIT	SELBSTSTUDIUM
Methoden und Werkzeuge in der Softwareentwicklung	30	45

- Einsatz von Entwicklungsumgebungen und Tools: Testen (z. B. JUnit), Debugging, Kommentieren und Dokumentieren.
- Grundlagen XML: Konfigurationsdateien und Datenaustauschformat.

LERNEINHEITEN UND INHALTE

LEHR- UND LERNEINHEITEN	PRÄSENZZEIT	SELBSTSTUDIUM
Moderne Programmierkonzepte	30	45

- Software-Ergonomie und Usability: Grundlagen, allgemeine Entwurfsprozesse und Entwurfsprinzipien für Benutzungsoberflächen, Interaktionsgestaltung, psychologische Aspekte, Navigation, praktische Umsetzung an Beispielen.
- Funktionale Programmierung: Grundlagen und praktische Umsetzung an Beispielen, Funktionen als Datenwerte.
- Streaming-Konzepte für Datenstrukturen: Grundlagen und praktische Umsetzung an Beispielen, Parallelisierungsaspekte.

BESONDERHEITEN

-

VORAUSSETZUNGEN

-

LITERATUR

- Beck, T.: Test Driven Development, By Example, Addison-Wesley.
- Kleuker, S.: Grundkurs Software-Engineering mit UML, Vieweg & Teubner Verlag, Wiesbaden.
- Koomen, T. und Pol, M.: Test Process Improvement, Addison-Wesley.
- Krug, S.: Don't Make Me Think: A Common Sense Approach to Web Usability, mitp Business.
- Künneth, Th.: Einstieg in Eclipse: Die Werkzeuge für Java-Entwickler, Galileo Computing.
- Liggesmeyer, P.: Software Qualität: Testen, Analysieren und Verifizieren von Software, Spektrum Akademischer Verlag.
- Ludewig, J.; Lichter, H.: Software Engineering: Grundlagen, Menschen, Prozesse, Techniken, dpunkt.Verlag, Heidelberg.
- Seacord, R. C., Plakosh, D. und Lewis, G. A.: Modernizing Legacy Systems, Addison-Wesley.
- Sommerville, I: Software Engineering, Addison-Wesley Longman.
- Subramaniam, V.: Functional Programming in Java: Harnessing the Power of Java 8 Lambda Expressions, O'Reilly.
- Thiemann, P.: Grundlagen der funktionalen Programmierung, Springer