

25 Softwaretechnik Software Engineering	
Semester	4
Dauer (Semester)	einsemestrig
Credit Points	5
Pflicht/ Wahlpflicht	Pflicht
Häufigkeit des Angebotes/ Verwendbarkeit	Im Aufnahmerhythmus
Modulverantwortliche(r)	Prof. Dr. Stefan Edlich, Beuth Hochschule für Technik Berlin
Lerngebiet	Informatik
Teilnahmevoraussetzungen	Sichere Anwendung von Hochsprachen wie Java, C#, etc.
Erwartungen	Sichere Anwendung von Hochsprachen wie Java, C#
Lernergebnisse	<p>Nach dem erfolgreichen Abschluss des Studienmoduls, sind die Studierenden in der Lage:</p> <ul style="list-style-type: none"> • softwaretechnische Kenntnisse in Projekte und in die Projektarbeit zu übertragen und anzuwenden • Anforderungsermittlung und Verwaltung eigenständig durchzuführen • informationstechnische Sachverhalte grafisch darzustellen • tragfähige IT-Architekturen zu entwerfen und zu gestalten • zu entscheiden und abzuwägen, wann welches (bestimmtes) Vorgehensmodell besser geeignet ist als ein anderes • Requirements Engineering im Rahmen der Projektarbeit einzusetzen und zu erklären. • die Hauptprobleme der Softwareentwicklung durch Analyse und Berücksichtigung der wichtigsten Anforderungsmerkmale zu identifizieren. • im Rahmen der Analyse - Pflichten- und Lastenheft, Use-Cases und Requirements einzuordnen und zu erstellen. • den geeigneten Einsatz von UML zu beurteilen und UML praktisch an einem eigenen Projekt anzuwenden und die kritische Nutzung dieser Industriesprache zu berücksichtigen. • zu beurteilen welche UML-Diagramme in welcher Reihenfolge anzuwenden sind, um ein Modellierungsziel zu erreichen • die Bedeutung der Architektur im Designprozess zu erklären und diese auf Projekte anzuwenden und zu begründen • Werkzeuge für das systematische und objektorientierte Testen einzusetzen und selber Tests zu entwerfen

	<ul style="list-style-type: none"> • die Möglichkeiten und Grenzen des Refactoring zu erklären und unter Eclipse oder einer anderen IDE anzuwenden, u.a. durch identifizieren von Bad Code Smell • die Funktionen des Buildmanagements mit ANT praktisch einzusetzen • die Konzepte des Versions- und Fehlermanagements zu erklären und die bekanntesten Systeme praxisnah zu verwenden • die Bedeutung von Metriken als Qualitätsmaß praktisch zu beurteilen und Basismetriken zu berechnen • Codemetriken und deren Werkzeuge zu gebrauchen, bspw. Architekturmetriken und deren Visualisierung • das Entwurfsmuster Dependency Injection unter Verwendung unterschiedlicher Frameworks in Projekten zu nutzen.
Prüfungsvorleistung	Pflicht-Präsenzteilnahme (12 x 45 Minuten)
Medien-/ Lernform	Multimedial aufbereitetes Online-Studienmodul zum Selbststudium mit zeitlich parallel laufender Online-Betreuung (E-Mail, Foren, Chat, Webkonferenzen, Einsendeaufgaben u. a.) sowie Präsenzphasen
Präsenzinhalte	<p>A) Praxisübungen mit UML. Durchführung eines konkreten Fallbeispiels</p> <p>B) Praxisübungen in den Bereichen Qualitätssicherung (Testen)</p> <p>C) Praxisübung in den Bereichen Buildmanagement, Versionsmanagement, etc.</p>
Prüfungsform	Klausur (120 min.)
Literatur	<p>Balzert, Lehrbuch der Softwaretechnik Oesterreich, Analyse und Design mit UML 2.5 Christ Rupp, Requirements Engineering Balzert, Lehrbuch der Objektmodellierung Ian Sommerville, Softwaretechnik (Global Edition)</p> <p>Jeckle, UML 2 glasklar</p>
weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Studieninhalte

LE01 Einführung in die Softwaretechnik
 LE02 Vorgehensmodelle / agile Modelle
 LE03 Requirements Engineering
 LE04 Analyse
 LE05 Unified Modeling Language
 LE06 Objektorientiertes Design
 LE07 Objektorientierte Architekturen
 LE08 Objektorientiertes Testen und Test-Driven Development
 LE09 Refactoring

LE10 Buildmanagement
LE11 Versions- und Fehlermanagement
LE12 Software- und Architekturmetriken
LE13 Dependency Injection