

4 Moderne Softwareentwicklung Modern Software Development	
Semester	1
Credit Points	5
Pflicht/ Wahlpflicht	Pflicht
Häufigkeit des Angebotes/ Verwendbarkeit	Jedes Semester nach Bedarf der Hochschulen im VFH-Verbund
Modulverantwortliche(r)	Prof. Dr. Stefan Edlich, Beuth Hochschule für Technik Berlin
Lerngebiet	Informatik
Teilnahmevoraussetzungen	empfohlen: Vorkenntnisse aus dem Bachelor-Modul Softwaretechnik
Lernergebnisse	<p>Festigung der Kenntnisse aus dem Bachelormodul Softwaretechnik und Erlernen neuer Methoden der Praxis.</p> <p>Nach dem erfolgreichen Abschluss des Studienmoduls, sind die Studierenden in der Lage:</p> <ul style="list-style-type: none"> • Elemente und Anwendung des UML Designs zu verstehen • ein praktisches Design zu entwerfen • Anforderungen und Systeme für die UML-Umsetzung zu analysieren • aus UML-Basiselementen ein System zu entwickeln • ihre vertieften Kenntnisse der Softwaretechnik, insbesondere modernere handwerkliche Fähigkeiten darzustellen und anzuwenden • moderne Techniken anzuwenden (Git, Jenkins, funktionale / loschige Prog., Maven / Gradle, Clean Code, etc.) • Moderne Tool-Chains in der Industrie wie praktisches Testen, Continuous Delivery, Coding Guidelines anzuwenden und DSLs zu entwickeln • reale Problemstellungen auf Werkzeuge abzubilden • aus den Bausteinen (z. B. der logischen oder funktionalen Programmierung) ein Gesamtsystem zu entwickeln • die Effizienz eines Gesamtsystems zu evaluieren und zu bewerten • Basiselemente der modernen Vorgehensweise einzusetzen • Elemente (z. B. composition in logischer Programmierung) zusammenzusetzen • praktische Übungen der Methodik durchzuführen wie z. B. das Ausschöpfen der Mächtigkeit von composition oder lazyness • Problemstellungen auf abstrakte Konzepte abzubilden (auch hier z. B. comp und lazy) • aus den Methoden (z. B. der logischen oder funktionalen Programmierung) den gewünschten Effekt zu erzielen • die Unterschiede der verschiedenen Ansätze zu bewerten, z. B. verschiedene Methoden der Sprachen

Prüfungsvorleistung	Einsendeaufgabe
Medien-/ Lernform	Multimedial aufbereitetes Online-Studienmodul zum Selbststudium mit zeitlich parallel laufender Online-Betreuung (E-Mail, Foren, Chat, Webkonferenzen, Einsendeaufgaben u. a.)
Arbeitsaufwand	Prüfung: 120 Minuten Selbststudium: 97,5 h Betreutes Lernen: 32,5 h Vorbereitung PVL: 20 h
Präsenzinhalte	Klärung inhaltlicher Fragen, gemeinsame Bearbeitung von Aufgaben und Übungen.
Prüfungsform	Klausur (120 min.)
Literatur	Oesterreich, Analyse und Design mit UML 2.3: Objektorientierte Softwareentwicklung, Oldenbourg Rady, Coffin, Continuous Testing with Ruby, The Pragmatic Bookshelf Noel Rappin, Rails Test Prescriptions, The Pragmatic Bookshelf Robert C. Martin, Clean Code, Prentice Hall Boris Gloger, Scrum, Hanser Verlag Jez Humble, Continuous Delivery, Addison-Wesley Signature Series Pastor, Model-Driven Architecture in Practice, Springer Fowler, Domain Specific Languages, Addison-Wesley Professional (Signature Series)
Eingangszweige	Medieninformatik, Medien
weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Studieninhalte

1. Unified Modeling Language
2. Advanced Java
3. Software- und Architekturmetriken
4. Clean Code Development
5. Advanced Buildmanagement
6. Distributed Version Control Systems Vertiefung
7. Continuous Integration / Continuous Delivery
8. Aspect Oriented Programming
9. Funktionale Programmierung
10. Logische Programmierung
11. Externe und interne DSLs