

1. Semester: Verfahren und Werkzeuge moderner Softwareentwicklung

Modul-Bezeichnung MSc Curriculum 2012	überarbeitet: Mündemann, 20.4.13	Stufen nach Bloom	Verfahren und Werkzeuge moderner Softwareentwicklung
englische Modulbezeichnung			Procedures and tools of modern Software Engineering
Abkürzung			VWS
Beschreibung erstellt	am		03.04.2013
	durch		Edlich, Mündemann
Niveaustufe			Master
Studiensemester			1
Kreditpunkte			5
Status	Pflichtmodul		Pflichtmodul
	Wahlpflichtmodul		
	Wahlmodul		
Häufigkeit des Angebotes			jedes Semester nach Bedarf der Hochschulen des VFH- Verbundes
Lehrsprache			Deutsch
Autoren			Prof. Dr. Edlich, Prof. Dr. Dragan Macos (Beuth-Hs Berlin)
Verantwortliche Hochschule			Beuth Hochschule Berlin
Fachverbands- leiter(in)			Prof. Dr. Edlich (Beuth-Hs Berlin)
Verantwortliche(r) Lehrende(r) am Standort nach dem Meister- Geselle-Prinzip: ein Lehrender für alle Standorte	Beuth-Hs Berlin		Prof. Dr. Edlich (Beuth-Hs Berlin)
	FH Brandenburg		Prof. Dr. Edlich (Beuth-Hs Berlin)
	FH Emden / Leer		Prof. Dr. Edlich (Beuth-Hs Berlin)
	FH Lünebeck		Prof. Dr. Edlich (Beuth-Hs Berlin)
	Ostfalia Hochschule Wolfenbüttel		Prof. Dr. Edlich (Beuth-Hs Berlin)
Lerngebiet			Informatik
Erworbene Kenntnisse, Fertigkeiten, Kompetenzen	Formale, algorithmische, mathematische Kompetenzen	Wissen	
		Verstehen	
		Anwenden	
		Analysieren	
		Synthetisieren	
		Evaluieren, Bewerten	
	Analyse-, Design- und Realisierungs- Kompetenzen	Wissen	Festigung der Kenntnisse aus SWT Bachelor (z.B. UML)
			neue Methoden der Praxis
		Verstehen	
		Anwenden	
		Analysieren	
		Synthetisieren	
	Technologische Kompetenzen	Wissen	Die Studenten vertiefen Kenntnisse der Softwaretechnik, insbesondere modernere handwerkliche Fähigkeiten
		Verstehen	
		Anwenden	Moderne Tool-Chains in der Industrie wie praktisches Testen, Continuous Delivery, Coding Guidelines anwenden, DSLs entwickeln, etc.
		Analysieren	
		Synthetisieren	
		Evaluieren, Bewerten	

	Fachübergreifende Kompetenzen	Wissen	Bezüge zum Fach Programmierung setzen
		Verstehen	
		Anwenden	
		Analysieren	
		Synthetisieren	
		Evaluiieren, Bewerten	
	Methodenkompetenzen	Wissen	
		Verstehen	
		Anwenden	
		Analysieren	
		Synthetisieren	
		Evaluiieren, Bewerten	
	Projektmanagement-Kompetenz	Wissen	
		Verstehen	
		Anwenden	
		Analysieren	
		Synthetisieren	
		Evaluiieren, Bewerten	
	Soziale Kompetenz und Selbstkompetenz	Wissen	
		Verstehen	
Anwenden			
Analysieren			
Synthetisieren			
Evaluiieren, Bewerten			
Obligatorische Teilnahmevoraussetzungen (nach Prüfungsordnung)			Keine
Fakultative Teilnahmevoraussetzungen			Softwaretechnik Bachelor
Medien-/Lernform			Multimedial aufbereitetes Online-Studienmodul zum Selbststudium mit zeitlich parallel laufender Online-Betreuung (E-Mail, Chat, Einsendeaufgaben u. a.) sowie Präsenzphasen (s. o.)
Arbeitsaufwand / work load (jeweils in Zeitstunden summiert)	Pflicht-Präsenzstudium	Vorlesung, Übung, Labor, Seminar u.a.	
		Modulprüfung	2h
	Online-Studium (Chat, Audio- / Videokonf. u.a.) incl. studentische Lerngruppen und fakultative Präsenzen		32h
	Erarbeiten der Prüfungsvorleistungen		16h
	Eigenstudium einschl. Prüfungsvorbereitung		100h
Summe Workload in Stunden			150h
Präsenzinhalte			Klärung inhaltlicher Fragen
Präsenzart	obligatorisch		
	fakultativ		fakultativ
Präsenzteilnahme	erfordert physische Anwesenheit		
	per web-Konferenz möglich		Die Vermittlung der Präsenzinhalte ist per Webkonferenz möglich.

Prüfungsvorleistungen	Präsenzteilnahme		
	Online-Teilnahme		
	Einsendeaufgaben		Mindestpunktzahl an Einsendeaufgaben (mind. 50%)
	Hausarbeit / Projektarbeit		
Teilleistungsnachweise			
Prüfungsform	Klausur	(120 Min)	Klausur (120 min)
	Mündliche Prüfung	(30 Min)	
	Hausarbeit mit Kolloquium	(30 Min)	
Benotung			Die Note ergibt sich aus der Abschlussklausur und den Übungen anteilig
Literatur			<p>Oesterreich, Analyse und Design mit UML 2.3: Objektorientierte Softwareentwicklung, Oldenbourg</p> <p>Rady, Coffin, Continuous Testing with Ruby, The Pragmatic Bookshelf</p> <p>Noel Rappin, Rails Test Prescriptions, The Pragmatic Bookshelf</p> <p>Robert C. Martin, Clean Code, Prentice Hall</p> <p>Boris Gloger, Scrum, Hanser Verlag</p> <p>Jez Humble, Continuous Delivery, Addison-Wesley Signature Series</p> <p>Pastor, Model-Driven Architecture in Practice, Springer</p> <p>Fowler, Domain Specific Languages, Addison-Wesley Professional (Signature Series)</p> <p>Debashish Gosh, "DSLs in Action", Manning,</p> <p>Martin Fowler, "Domain Specific Languages", Addison Wesley Longman</p>
Weitere Hinweise			
Studieninhalte des Moduls (Allgemeines zum Modul / Zusammenfassung)			
Kapitelüberschriften / Überschriften der Lerneinheiten			LE 01 UML Vertiefung
			LE 02 Agile: Scrum und Praxis
			LE 03 MDA / MDDD Standards, Formate & Best Practices
			LE 04 Clean Code Development
			LE 05 Fortgeschrittenes Testen (Behaviour Driven)
			LE 06 AOP Aspekt Oriented Programming
			LE 07 DSLs
			LE 08 Advanced Buildmanagement
			LE 09 Continuous Integration / Delivery
			LE 10 DVCS Distributed Version Control Systems
			LE 11 Advanced Java
			LE 12 Programmierparadigmen (Paradigmen in Clojure, Scala, Prolog, Haskell, Erlang, etc.)