

Modul: Softwaretechnik

Niveau	Bachelor	Stundenplankürzel	SWT
Modulname englisch	Software Engineering		
Modulverantwortliche	Prof. Dr. Stefan Krause		
Fachbereich	Elektrotechnik und Informatik		
Studiengang	Informationstechnologie und Design, Bachelor		
Verpflichtungsgrad	Pflicht	ECTS-Leistungspunkte	12
Fachsemester	2	Semesterwochenstunden	6
Dauer in Semestern	2	Arbeitsaufwand in Stunden	360
Angebotshäufigkeit	SoSe	Präsenzstunden	90
Lehrsprache	Deutsch	Selbststudiumsstunden	270

Der folgende Abschnitt ist nur ausgefüllt, wenn es **genau eine** modulabschließende Prüfung gibt.

Prüfungsleistung		Prüfsprache	
Dauer PL in Minuten		Bewertungssystem PL	
Lernergebnisse			
Teilnahmevoraussetzungen			

Der vorige Abschnitt ist nur ausgefüllt, wenn es **genau eine** modulabschließende Prüfung gibt.

Berücksichtigung von Gender- und Diversity-Aspekten	<ul style="list-style-type: none"> ✓ Verwendung geschlechtergerechter Sprache (THL-Standard) ✓ Zielgruppengerechte Anpassung der didaktischen Methoden ✗ Sichtbarmachen von Vielfalt im Fach (Forscherinnen, Kulturen etc.)
Verwendbarkeit	
Bemerkungen	<p>2. Semester (Teil 1):</p> <p>2 V mit integrierten Übungen</p> <p>2 Pr Gruppenarbeit (6-8 Studierende pro Gruppe)</p> <p>Begleitete, selbständige Durchführung eines Projekts zum systematischen Entwurf eines Softwaremodells für eine gegebene oder selbst gewählte Aufgabenstellung.</p> <p>3. Semester (Teil 2):</p> <p>2 Pr Gruppenarbeit (6-8 Studierende pro Gruppe)</p> <p>Begleitete, selbständige Durchführung eines Projekts zur softwaretechnischen Realisierung (Programmierung, Test) des im ersten Teil des Moduls erarbeiteten Softwaremodells (siehe oben). Typischerweise bleiben die im ersten Teil des Moduls formierten Gruppen in ihrer Zusammensetzung erhalten.</p>

Bei der Durchführung des Praktikums ist die vorgesehene Reihenfolge (erst Teil 1, dann Teil 2) einzuhalten.

Teilaufgaben sind individuell festzulegen/durchzuführen und gemeinsam zusammen zu führen. Unterstützende Vorträge durch Dozenten.

Lehrveranstaltung: Softwaretechnik

(zu Modul: Softwaretechnik)

Lehrveranstaltungsart	Vorlesung	Lernform	Präsenz
LV-Name englisch	Software Engineering		
Anwesenheitspflicht	nein	ECTS-Leistungspunkte	4
Teilnahmebeschränkung		Semesterwochenstunden	2
Gruppengröße		Arbeitsaufwand in Stunden	120
Lehrsprache	Deutsch	Präsenzstunden	30
Studienleistung		Selbststudiumsstunden	90
Dauer SL in Minuten		Bewertungssystem SL	

Der folgende Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Prüfungsleistung	Klausur	Prüfsprache	Deutsch
Dauer PL in Minuten	90	Bewertungssystem PL	Drittelnoten

Lernergebnisse	<p>Aufbauend auf Grundkenntnissen der (objektorientierten) Programmierung verstehen die Studierenden nach dem Studium dieses Moduls, welche Bedeutung eine ingenieur-mäßige Herangehensweise für die Entwicklung großer Softwaresysteme hat. Sie lernen theoretische Grundlagen und Zusammenhänge kennen, um übergreifende fachliche Problemstellungen zu verstehen und um neuere technisch wissenschaftliche Entwicklungen einordnen, verfolgen und mitgestalten zu können. Durch die interaktive Bearbeitung eines durchgängigen Fallbeispiels in der Vorlesung und die teambasierte Durchführung eines eigenständigen Softwareprojekts im Praktikum verbessern die Studierenden gleichzeitig ihre logisch analytische Denkweise, ihre Problemlösungskompetenz sowie ihre Teamfähigkeit.</p> <p>Die Studierenden werden nach Abschluss des Moduls in der Lage sein, von einer gegebenen Problemstellung systematisch zu einer lauffähigen Softwarelösung zu gelangen. Dies betrifft sämtliche Phasen des sog. Software-„Lebenszyklus“, von der Problem- und Anforderungsanalyse über den softwaretechnischen Entwurf (inkl. Prototyping), die Programmierung, das Testen und die Inbetriebnahme bis zum Re-Engineering. Dabei stehen die objektorientierten Methoden der Softwareentwicklung im Mittelpunkt, unterstützt von UML als Modellierungsnotation und einer objektorientierten Programmiersprache wie Java. Die Studierenden lernen, Werkzeuge, Frameworks und Funktionsbibliotheken einzusetzen, ein Softwareprojekt zu organisieren, im Team zu arbeiten sowie Projektergebnisse überzeugend zu präsentieren.</p>
Teilnahmevoraussetzungen	Kenntnisse in objektorientierter Programmierung (Java)

Der vorige Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Lehrinhalte	Grundlagen der Softwaretechnik (Vorlesung) <ul style="list-style-type: none"> Anforderungen an Software
--------------------	--

- Softwaretechnik
 - Einordnung
 - Vorgehensmodelle

Software-Lebenszyklus (Vorlesung und Praktikum)

- Problemanalyse (Lastenheft)
- Anforderungsdefinition
 - Pflichtenheft
 - Objektorientierte Analyse (OOA, UML)
 - Anwendungsfalldiagramme
 - Aktivitätsdiagramme
 - Klassendiagramme (Fachklassen)
 - Sequenzdiagramme
- Modellierung der Benutzungsschnittstelle
- Entwurf
 - Grobentwurf (Software-Architektur)
 - Feinentwurf (OOD, UML)
 - Klassendiagramme (Control- und Boundary-Klassen)
 - Sequenzdiagramme
- Implementierung (objektorientierte Programmierung)
 - Entwicklungsumgebungen
 - Anbindung von Datenbanksystemen
 - Einsatz von GUI-Bibliotheken
 - Code-Dokumentation
 - Integration
- Test
 - Statische Verfahren (Code Review)
 - Dynamische Verfahren
 - Testwerkzeuge
- Ergebnispräsentation und Abnahme

Softwareprojektmanagement (Vorlesung und Praktikum)

- Konfigurations- und Versionsmanagement
- Termin-, Personal-, Kommunikationsmanagement

Literatur

Balzert, Heide: Lehrbuch der Objektmodellierung: Analyse und Entwurf mit der UML 2, Spektrum 2005

Balzert, Helmut: Lehrbuch der Software-Technik - Entwurf, Implementierung, Installation und Betrieb, Spektrum 2011

Brügge, Bernd; Dutoit, Allen H.: Object-Oriented Software Engineering Using UML, Patterns, and Java (3rd edition), Prentice Hall 2010, oder auch:

Brügge, Bernd; Dutoit, Allen H.: Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java, Pearson Studium, 2004

Ian Sommerville: Software Engineering, 9th ed., Pearson 2011

UML Superstructure Specification v2.4.1; <http://www.omg.org/spec/UML/2.4.1/Superstructure>

Spillner, A.; Linz, T.: Basiswissen Softwaretest, dpunkt.verlag, 4. Auflage, 2010

Vigenschow, Uwe: Objektorientiertes Testen und Testautomatisierung in der Praxis, dpunkt 2004

Bemerkungen

Lehrveranstaltung: Softwaretechnik Praktikum

(zu Modul: Softwaretechnik)

Lehrveranstaltungsart	Praktikum	Lernform	Präsenz
LV-Name englisch	Software Engineering Practical Training		
Anwesenheitspflicht	ja	ECTS-Leistungspunkte	8
Teilnahmebeschränkung		Semesterwochenstunden	4
Gruppengröße	12	Arbeitsaufwand in Stunden	240
Lehrsprache	Deutsch	Präsenzstunden	60
Studienleistung		Selbststudiumsstunden	180
Dauer SL in Minuten		Bewertungssystem SL	

Der folgende Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Prüfungsleistung	Projektarbeit	Prüfsprache	Deutsch
Dauer PL in Minuten		Bewertungssystem PL	Drittelnoten
Lernergebnisse	Siehe Vorlesung. Zusätzlich lernen die Studierenden, Werkzeuge, Frameworks und Funktionsbibliotheken einzusetzen, ein Softwareprojekt zu organisieren, im Team zu arbeiten sowie Projektergebnisse überzeugend zu präsentieren.		
Teilnahmevoraussetzungen	Kenntnisse in objektorientierter Programmierung (Java)		

Der vorige Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Lehrinhalte	Vertiefung und praktische Anwendung der in der Vorlesung erworbenen Kenntnisse
Literatur	<p>Balzert, Heide: Lehrbuch der Objektmodellierung: Analyse und Entwurf mit der UML 2, Spektrum 2005</p> <p>Balzert, Helmut: Lehrbuch der Software-Technik - Entwurf, Implementierung, Installation und Betrieb, Spektrum 2011</p> <p>Brügge, Bernd; Dutoit, Allen H.: Object-Oriented Software Engineering Using UML, Patterns, and Java (3rd edition), Prentice Hall 2010, oder auch:</p> <p>Brügge, Bernd; Dutoit, Allen H.: Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java, Pearson Studium, 2004</p> <p>Ian Sommerville: Software Engineering, 9th ed., Pearson 2011</p> <p>UML Superstructure Specification v2.4.1; http://www.omg.org/spec/UML/2.4.1/Superstructure</p> <p>Spillner, A.; Linz, T.: Basiswissen Softwaretest, dpunkt.verlag, 4. Auflage, 2010</p>

	Vigenschow, Uwe: Objektorientiertes Testen und Testautomatisierung in der Praxis, dpunkt 2004
Bemerkungen	<p>Die Gesamtnote des Moduls setzt sich aus folgenden Teilprüfungsleistungen zusammen, die unabhängig voneinander erbracht werden können:</p> <p>Klausur (90 Minuten), Gewicht 1/3</p> <p>Projektarbeit, Gewicht 2/3</p> <p>Die Projektarbeit erstreckt sich über beide Semester. Bewertet werden Art der Durchführung, Engagement, Erfolg, Zusammenarbeit und Darstellung. Die Note für die Projektarbeit setzt sich wie folgt zusammen:</p> <p>Modellierung (2. Sem.), Gewicht 1/3</p> <p>Realisierung (3. Sem.), Gewicht 2/3</p> <p>Begründung für separate Projektarbeit: Wesentlicher Lerninhalt ist die praktische Umsetzung der behandelten Lösungsansätze. Diese lassen sich nur im Rahmen einer semesterbegleitenden Projektarbeit überprüfen. Die Klausurinhalte prüfen lediglich die in der Vorlesung vermittelten theoretischen Konzepte ab und nicht die im Praktikum erworbenen Fertigkeiten. Die Berechnung der Workload beruht auf dieser inhaltlichen Trennung.</p>