

Modul: Softwaretechnik

Niveau	Bachelor	Kürzel	SWT
Modulname englisch	Software Engineering		
Modulverantwortliche	Krause, Stefan, Prof. Dr.		
Fachbereich	Elektrotechnik und Informatik		
Studiengang	Informationstechnologie und Design, Bachelor		
Verpflichtungsgrad	Pflicht	ECTS-Leistungspunkte	7
Fachsemester	2	Semesterwochenstunden	4
Dauer in Semestern	1	Arbeitsaufwand in Stunden	210
Angebotshäufigkeit	SoSe	Präsenzstunden	60
Lehrsprache	Deutsch	Selbststudiumsstunden	150

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	2 V mit integrierten Übungen 2 Pr Gruppenarbeit (ca. 3-4 Studierende pro Gruppe) Begleitete, selbständige Durchführung eines Projekts zur Anforderungsanalyse und zum systematischen Entwurf eines Softwaremodells für eine gegebene oder selbst gewählte Aufgabenstellung.

Lehrveranstaltung: Softwaretechnik (Vorlesung)

(zu Modul: Softwaretechnik)

Lehrveranstaltungsart	Vorlesung	Lernform	Präsenz
LV-Name englisch	Software Engineering (Lecture)		
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	Drittelnoten

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

Prüfungsleistung	Klausur	Prüfungsprache	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 ingenieurmäßige 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 Problemstellung systematisch zu einer Anforderungsanalyse und einem technischen Entwurf zu gelangen. Dabei stehen die objektorientierten Methoden der Softwareentwicklung im Mittelpunkt, unterstützt von UML als Modellierungsnotation.</p>
Teilnahmevoraussetzungen	Grundkenntnisse der Programmierung (Java)

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

Lehrinhalte	<p>Grundlagen der Softwaretechnik (Vorlesung)</p> <ul style="list-style-type: none"> • Anforderungen an Software • Softwaretechnik <ul style="list-style-type: none"> • Einordnung • Vorgehensmodelle <p>Software-Lebenszyklus (Vorlesung und Praktikum)</p> <ul style="list-style-type: none"> • Problemanalyse (Lastenheft)
--------------------	---

	<ul style="list-style-type: none"> • Anforderungsdefinition <ul style="list-style-type: none"> • Pflichtenheft • Objektorientierte Analyse (OOA, UML) <ul style="list-style-type: none"> • Anwendungsfalldiagramme • Aktivitätsdiagramme • Klassendiagramme (Fachklassen) • Sequenzdiagramme • Modellierung der Benutzungsschnittstelle • Entwurf <ul style="list-style-type: none"> • Grobentwurf (Software-Architektur) • Feinentwurf (OOD, UML) <ul style="list-style-type: none"> • Klassendiagramme (Control- und Boundary-Klassen) • Sequenzdiagramme • Implementierung (objektorientierte Programmierung) <ul style="list-style-type: none"> • Entwicklungsumgebungen • Anbindung von Datenbanksystemen • Einsatz von GUI-Bibliotheken • Code-Dokumentation • Integration • Test <ul style="list-style-type: none"> • Statische Verfahren (Code Review) • Dynamische Verfahren • Testwerkzeuge • Ergebnispräsentation und Abnahme <p>Softwareprojektmanagement (Vorlesung und Praktikum)</p> <ul style="list-style-type: none"> • Konfigurations- und Versionsmanagement • Termin-, Personal-, Kommunikationsmanagement
Literatur	<p>Ian Sommerville: Software Engineering, 10th ed., Pearson 2018</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>UML Superstructure Specification v2.4.1; http://www.omg.org/spec/UML/2.4.1/Superstructure</p> <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>Spillner, A.; Linz, T.: Basiswissen Softwaretest, dpunkt.verlag, 4. Auflage, 2010</p> <p>Vigenschow, Uwe: Objektorientiertes Testen und Testautomatisierung in der Praxis, dpunkt 2004</p>
Bemerkungen	

Lehrveranstaltung: Softwaretechnik (Praktikum)

(zu Modul: Softwaretechnik)

Lehrveranstaltungsart	Praktikum	Lernform	Präsenz
LV-Name englisch	Software Engineering Practical Training		
Anwesenheitspflicht	ja	ECTS-Leistungspunkte	3
Teilnahmebeschränkung		Semesterwochenstunden	2
Gruppengröße	12	Arbeitsaufwand in Stunden	90
Lehrsprache	Deutsch	Präsenzstunden	30
Studienleistung		Selbststudiumsstunden	60
Dauer SL in Minuten		Bewertungssystem SL	Teilnahme

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

Prüfungsleistung	Studienarbeit	Prüfungsprache	Deutsch
Dauer PL in Minuten		Bewertungssystem PL	Bestehen
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	Siehe Vorlesung
Bemerkungen	