

**Modul: Softwaretechnik**

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

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

<b>Prüfungsleistung</b>		<b>Prüfsprache</b>	
<b>Dauer PL in Minuten</b>		<b>Bewertungssystem PL</b>	
<b>Lernergebnisse</b>			
<b>Teilnahmevoraussetzungen</b>			

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

<b>Berücksichtigung von Gender- und Diversity-Aspekten</b>	<ul style="list-style-type: none"> <li>✓ Verwendung geschlechtergerechter Sprache (THL-Standard)</li> <li>✓ Zielgruppengerechte Anpassung der didaktischen Methoden</li> <li>✗ Sichtbarmachen von Vielfalt im Fach (Forscherinnen, Kulturen etc.)</li> </ul>
<b>Verwendbarkeit</b>	
<b>Bemerkungen</b>	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)

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

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

<b>Prüfungsleistung</b>	Klausur	<b>Prüfsprache</b>	Deutsch
<b>Dauer PL in Minuten</b>	90	<b>Bewertungssystem PL</b>	Drittelnoten

<b>Lernergebnisse</b>	<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>
<b>Teilnahmevoraussetzungen</b>	Grundkenntnisse der Programmierung (Java)

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

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

- 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**

Ian Sommerville: Software Engineering, 10th ed., Pearson 2018

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

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

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

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)

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

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

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

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

<b>Lehrinhalte</b>	Vertiefung und praktische Anwendung der in der Vorlesung erworbenen Kenntnisse.
<b>Literatur</b>	Siehe Vorlesung
<b>Bemerkungen</b>	