Modulbezeichnung	**2.4. Softwaretechnik**
Kürzel für Stundenplan	SwT
Semester	2 und 3
Modulverantwortliche(r)	Prof. Dr.-Ing. S. Krause
Dozent(in)	Prof. Dr.-Ing. S. Krause, Prof. Dr. H. Hinrichs
Sprache	Deutsch
Zuordnung zum Curriculum	Pflichtmodul
Lehrform / SWS	2. Semester (Teil 1): 2 V mit integrierten Übungen 2 Pr Gruppenarbeit (6-8 Studierende pro Gruppe) Begleitete, selbständige Durchführung eines Projekts zum systematischen Entwurf eines Softwaremodells für eine gegebene oder selbst gewählte Aufgabenstellung. 3. Semester (Teil 2): 1 Pr Gruppenarbeit (6-8 Studierende pro Gruppe) 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. 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.
Arbeitsaufwand	2. Semester (Teil 1): 64 h Präsenz (32h Vorlesung, 32h Praktikum) 48 h Vor-/Nachbereitung der Vorlesung und Übungen 48 h Vor-/Nachbereitung des Praktikums = 160 h 3. Semester (Teil 2): 160 h Projektbearbeitung, davon 16 h wöchentliche Besprechungen mit dem/der Lehrenden 12 h Vorbereitung der Abschlusspräsentation = 320 h
Kreditpunkte (gem. ECTS)	12 (6+6)
Voraussetzungen	Kenntnisse in objektorientierter Programmierung

Modulbezeichnung	**2.4. Softwaretechnik**
Lernziele / Kompetenzen	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. 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.

Modulbezeichnung	**2.4. Softwaretechnik**

Inhalt	Grundlagen der Softwaretechnik (Vorlesung)
	• 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
	  • Schichtenarchitekturen
	  • Client/Server-Modell
	  • 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 (funktionsorientierte und strukturorientierte)
	  • Testwerkzeuge
	• Ergebnispräsentation und Abnahme
	
	<u>Softwareprojektmanagement (Vorlesung und Praktikum)</u>
	• Konfigurations- und Versionsmanagement
	• Termin- , Personal-, Kommunikationsmanagement

Modulbezeichnung	**2.4. Softwaretechnik**
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 Oestereich, Bernd: Objektorientierte Softwareentwicklung, Oldenbourg 2005 Vigenschow, Uwe: Objektorientiertes Testen und Testautomatisierung in der Praxis, dpunkt 2004
Studien-/Prüfungsleistungen	Die Gesamtnote des Moduls setzt sich aus folgenden Teilprüfungsleistungen zusammen, die unabhängig voneinander erbracht werden können: Klausur (90 Minuten), Gewicht 1/3 Projektarbeit, Gewicht 2/3 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: Modellierung (2. Sem.), Gewicht 1/3 Realisierung (3. Sem.), Gewicht 2/3 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.