

Modulbezeichnung: Software-Entwicklung 1	
Modulkürzel	t.BA.IT.SWEN1.19HS
ECTS Credits	4
Unterrichts- und Prüfungssprache	Deutsch
Verantwortliche OE	InIT
Modulverantwortung	Kurt Bleisch
Rechtliche Grundlagen	Die Modulbeschreibung ist neben Rahmenprüfungsordnung und Studienordnung Teil der Rechtsgrundlage. Sie ist verbindlich. Eine in der ersten Unterrichtswoche des Semesters schriftlich festgehaltene und kommunizierte Modulvereinbarung kann die Modulbeschreibung präzisieren. Die Modulvereinbarung ersetzt nicht die Modulbeschreibung.
Modulprägung	Typ 2a 4 Lektionen Vorlesung aufeinanderfolgend pro Semesterwoche und Klasse
Beschreibung des Moduls	Dieses Modul vermittelt Ihnen die nötigen Analyse- und Entwurfs-Kompetenzen, um grössere und komplexere Softwareapplikationen entwickeln und realisieren zu können. Für die Analyse einer Problemstellung werden User und Domain Research, Use Cases und Domänenmodellierung angewendet. Die erhobenen Anforderungen an ein Softwaresystem werden dabei systematisch spezifiziert und geprüft. Abgeleitet von den Anforderungen werden Techniken zum Entwurf einer geeigneten Softwarearchitektur und der Umsetzung in ein objektorientiertes Design vermittelt. Dabei werden bewährte Architektur und Design Patterns angewendet und es wird auf eine hohe Softwarequalität (Erweiterbarkeit, Wartbarkeit) Wert gelegt. Für die Modellierung und Kommunikation der Resultate aus der Analyse, der Softwarearchitektur und dem Design werden standardisierte Notationen (wie UML) verwendet.

Modulbezeichnung: Software-Entwicklung 1

Inhalte des Moduls

Dieses Modul vermittelt die nötigen Analyse- und Entwurfs-Kompetenzen, um grössere und komplexere Softwareapplikationen entwickeln und realisieren zu können.

Zur Analyse-Kompetenz gehören vor allem der Wille und die Fähigkeit, mit Aufgabenstellern und zukünftigen Systemnutzern zu kommunizieren und zu kooperieren und sich schnell in neue Anwendungskontexte einarbeiten zu können. Die Studierenden müssen bekannte Problemstellungen im Anwendungskontext erkennen können und mit den zugehörigen Lösungsmustern vertraut sein. Sie erkennen Inkonsistenzen und können mit unklaren Anforderungen umgehen. Komplexe Domänen können modelliert werden und grosse Anwendungsprobleme können durch geeignete Schnittstellen in Teilprobleme zerlegt werden.

Entwurfs-Kompetenzen umfassen die Fähigkeit zur Konstruktion von Systemen aus Hard- und Software, welche die Anforderungen vollständig erfüllen. Hierfür ist Abstraktionsfähigkeit genauso unverzichtbar wie solide Kenntnisse in der Softwarearchitektur. Zentral ist beim Entwurf die Umsetzung nicht funktionaler Anforderungen, wie Sicherheit, Performanz, Skalierbarkeit, Wartbarkeit, Erweiterbarkeit und Zuverlässigkeit.

Einführung und Softwareentwicklungsprozessmodelle (4 Lektionen)

Überblick über Softwareentwicklungsprozessmodelle und deren Anwendungsgebiete (Wasserfall, iterativ-inkrementell und agile)

Ablauf und Artefakte in einem iterativ-inkrementellen, Use-Case-getriebenen und architekturzentrierten Softwareentwicklungsprozess

Anforderungsanalyse (16 Lektionen)

Einführung in Usability und UX (Contextual Inquiry, Personas und Szenarien, UI-Sketching & Prototyping)

Erhebung und Kommunikation von funktionalen Anforderungen mit Use Cases (UML-Use-Case-Diagramm, Use-Case-Spezifikation)

Erhebung und Kommunikation von nicht funktionalen Anforderungen (Qualitätsanforderungen, Randbedingungen)

Modellierung der Fachlichkeit und Begriffe des Anwenders (Domänenmodell) und Einführung in Domain Driven Design (DDD, konzeptuelles UML-Klassendiagramm)

Softwarearchitektur und Design (36 Lektionen)

Entwurf und Modellierung einer Softwarearchitektur (4+1 Sicht, UML-Paketdiagramm, UML-Deploymentdiagramm)

Einführung in Clean Architecture (SOLID Prinzipien, Schichtenarchitektur, Onion Architecture)

Use-Case-Realisierung und Klassendesign (Responsibility Driven Design (RDD), UML-Klassendiagramm, UML-Sequenzdiagramm, UML-Kommunikationsdiagramm, UML-Zustandsdiagramm, UML-Aktivitätsdiagramm)

Entwurf mit Design Patterns (GoF: Factory, Singleton, Adaptor, Bridge, Composite, Decorator, Facade, Proxy, Chain of Responsibility, Observer, State, Strategy, Visitor)

Verschiedene Vertiefungsthemen wie zum Beispiel: Verteilte Systeme, GUI-Architekturen, Persistenz, Framework-Design

Praktikum

Im Praktikum lösen die Studierenden jeweils auf das Thema der Vorlesung abgestimmte Übungen. Die Praktika sind integraler Bestandteil der Vorlesung in

Modulbezeichnung: Software-Entwicklung 1 diesem Modultyp (jeweils 2 Lektionen Vorlesung und 2 Lektionen Übungen).

Vorkenntnisse -

Lernziele (Kompetenzen)	Die Studierenden...	Kompetenzen	Taxonomiestufen
	Sie können für die Modellierung und Kommunikation von Artefakten im Softwareentwicklungsprozess standardisierte Notationen (wie UML) benutzen.	F, M	K3
	Sie können eine Softwareapplikation abgrenzen und dafür systematisch die funktionalen Anforderungen mit Use Cases sowie Qualitätsanforderungen und Randbedingungen erheben und kommunizieren.	M, F	K3
	Sie können die Begriffswelt des Anwenders durch geeignete Vorgehensweisen erfassen und zu einer fachlichen Terminologie verdichten (Domänenmodell).	M, F	K3
	Sie können, basierend auf den Anforderungen, eine geeignete Softwarearchitektur und ein objektorientiertes Design für die darin enthaltenen Komponenten der fachlichen Logik (Domänenlogik) entwerfen.	F, M	K3
	Sie können für einen vorgegebenen, iterativ-inkrementellen Softwareentwicklungsprozess den Ablauf und die Artefakte zur Entwicklung einer objektorientierten Softwareapplikation erläutern.	F, M	K2
	Sie können bewährte Analyse, Architektur und Design Patterns adäquat für eine Problemstellung einsetzen.	M, F	K3, K4

Leistungsnachweis	Modulendprüfung	Bewertung	Dauer (Min.)	Gewichtung	Form
	schriftliche Prüfung	Note	90	70	gem. Modulvereinbarung
	Leistungsnachweise während dem Semester	Bewertung	Dauer (Min.)	Gewichtung	Form
	Quiz <i>Quiz mit MC Fragen</i>	Note	15	5	gem. Modulvereinbarung
	Quiz <i>Quiz mit MC Fragen</i>	Note	15	5	gem. Modulvereinbarung
	Bewertetes Praktikum	Note		20	gem. Modulvereinbarung

Präsenzverpflichtung im Kontaktstudium Keine

Lernmaterialien

Bemerkungen