

Modulbezeichnung: Software-Projekt 3	
Modulkürzel	t.BA.IT.PM3.19HS
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 4* 4 Lektionen Praktikum pro Semesterwoche und Halbkasse
Beschreibung des Moduls	In diesem Modul entwickeln Sie eine grössere objektorientierte Softwareapplikation. Dabei werden die vermittelten Analyse- und Entwurfs-Kompetenzen aus dem Modul SWEN1 angewendet (Wissenstransfer). Weiter werden Realisierungs- und Projektmanagement-Kompetenzen beübt sowie die Sozial- und Selbstkompetenzen gestärkt. Sie realisieren in Teams eine selbstgewählte Projektidee von der Vision bis zu einer lauffähigen Applikation. Es wird ein vorgegebener interaktiv-inkrementeller Softwareentwicklungsprozess angewendet (Basis agiler Softwareentwicklung). Das Softwaresystem wird inkrementell – in Schritten – umgesetzt, getestet und laufend verbessert (Refactoring). Es werden konsequent Analyse-, Architektur- und Design Patterns sowie angemessene Frameworks und Tools eingesetzt. Dabei wird der Softwarequalität ein hoher Wert beigemessen (Clean Architecture, Clean Code).
Inhalte des Moduls	<p>Die Studierenden entwickeln in Teams von 5 (+/-1) Personen selbständig eine grössere Softwareapplikation in diesem Modul. Neben der Anwendung von Analyse- und Entwurfs-Kompetenzen (aus Modul SWEN1) werden auch die Realisierungs- und Projektmanagement-Kompetenzen beübt sowie die Sozial- und Selbstkompetenzen gestärkt.</p> <p>Zur Realisierungs-Kompetenz gehört vor allem die Fähigkeit, professionell grössere Softwaresysteme zu erstellen und sorgfältig zu testen.</p> <p>Projektmanagement-Kompetenz wird benötigt, um die Arbeitsprozesse zu gestalten und insbesondere die eigene und anderer Personen Arbeit zu organisieren. Hierfür sind Teamfähigkeit und die konstruktive Auseinandersetzung mit Konzepten und Lösungsvorschlägen essentiell. Studierende haben gelernt, auch unter begrenzten Ressourcen Lösungen zu erarbeiten, die allgemein anerkannten Qualitätsstandards genügen und von allen Beteiligten akzeptiert werden.</p> <p>Die Teams werden ausgelost, wählen einen Projektleiter und entwickeln gemeinsam eine Projektidee, die sie realisieren wollen. Die Meilensteine und die zu erstellenden Entwicklungs- und Projektmanagement-Artefakte sind vorgegeben. Ansonsten organisiert sich das Team selbst.</p> <p>Die Studierenden verwenden dabei einen vorgegebenen iterativ-inkrementellen Softwareentwicklungsprozess (Basis agiler Softwareentwicklung) mit den folgenden vorgegebenen Meilensteinen und Artefakten.</p> <p>M1: Projektskizze</p> <p>1 Iteration (2 Wochen)</p> <p>Evaluation: Stakeholder-Agreement über Vision, Projektziele und Umfang, vereinbarter erster Phasen-/Iterationsplan, erste Risiken identifiziert.</p> <p>Artefakte: Projektskizze (Vision, Konkurrenzanalyse, erste Anforderungen, Wirtschaftlichkeit), Projektmanagement (Phasen-/Iterationsplan, Iterationsplan nächste Iteration, Risikoliste)</p>

Modulbezeichnung: Software-Projekt 3

naechste Iteration, Risikoliste),
M2: Lösungsarchitektur

3 Iterationen (4 - 6 Wochen)

Evaluation: Vision, Ziele und Anforderungen klar dokumentiert und stabil (80%), dokumentierte und verifizierte Softwarearchitektur (mit Code), die den Anforderungen gerecht wird.

Artefakte: Use-Case-Modell, Domänenmodell, nicht funktionale Anforderungen, Softwarearchitektur, ausgewählte Designs, Code, Projektmanagement (Phasen-/Iterationsplan, Iterationspläne, Iterations-Assessments, Risikoliste)

M3: Beta-Release

2 Iterationen (4 Wochen)

Evaluation: Feature-complete, Beta-Release ist stabil

Artefakte: Applikation (Software), Technischer Bericht (Zusammenfassung der Analyse, des Designs und der Implementation), Projektmanagement (Phasen-/Iterationsplan, Iterationspläne, Iterations-Assessments, Risikoliste)

In jeder Iteration (ausser der ersten Iteration) wird ein Software-Inkrement implementiert. Die Resultate der Meilensteine werden jeweils pro Team im Plenum präsentiert. Dazwischen finden wöchentliche Statusmeetings pro Team mit dem Fachdozenten statt.

Während des gesamten Projekts verwenden die Studierenden eine Versionsverwaltung (Git) für den Code sowie Build- und Test-Tools (JUnit, Mockito, Gradle o.ä.).

Vorkenntnisse

-

Lernziele (Kompetenzen)

Die Studierenden...	Kompetenzen	Taxonomiestufen
Sie können eine grösseres objektorientierte Softwareapplikation von der Vision bis zur lauffähigen Applikation mit einem vorgegebenen iterativ-inkrementellen Softwareentwicklungsprozess realisieren.	M, F	K3
Sie können ein objektorientiertes Design inkrementell umsetzen, testen und laufend verbessern (Refactoring).	M, F	K3, K4
Sie können eine technische Recherche (Konkurrenzanalyse, Technologie) durchführen und standardkonform festhalten.	F, M	K2, K3
Sie können die im Softwareentwicklungsprozess definierten Entwicklungs- und Projektmanagement-Artefakte selbständig erstellen.	M, F	K3
Sie können die erstellen Entwicklungs-Artefakte mit einer standardisierten Notation (wie UML) modellieren, darstellen und kommunizieren.	F, M	K3
Sie können konsequent Analyse-, Architektur- und Design Patterns sowie angemessen Frameworks und Tools (für Versionierung, Build etc.) einsetzen.	M, F	K3
Sie können das Projekt im Team iterativ und risikogetrieben planen und abwickeln.	F, M	K3
Sie können technische Inhalte adressatengerecht mündlich und schriftlich kommunizieren.	SO	K3

Modulbezeichnung: Software-Projekt 3

Leistungsnachweis	Modulendprüfung	Bewertung	Dauer (Min.)	Gewichtung	Form
	andere	Note		100	
	Leistungsnachweise während dem Semester	Bewertung	Dauer (Min.)	Gewichtung	Form
	-	-	-	-	-
Präsenzverpflichtung im Kontaktstudium	Keine				
Lernmaterialien					
Bemerkungen					