



Modul Software-Entwicklung 1 (SWEN1)

Informationsblatt

1 Allgemeines

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

Für die Analyse einer Problemstellung werden User Research, Use Cases und Domänenmodellierung angewendet. Die erhobenen Anforderungen an eine Softwareapplikation werden dabei systematisch spezifiziert und geprüft.

Abgeleitet von den Anforderungen werden Methoden und Techniken zum Entwurf einer geeigneten Softwarearchitektur und der Umsetzung in ein objektorientiertes Design vermittelt. Dabei werden bewährte Architektur und Design Patterns angewendet und 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.

Das Modul Software-Projekt 3 (PM3) ist mit SWEN1 abgestimmt, so dass das Gelernte in einem eigenen grösseren Projekt angewendet werden kann (Wissenstransfer).

2 Lernziele

Sie sind in der Lage,

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



3 Unterrichtsplan

Der folgende Unterrichtsplan bzw. Wochenplan zeigt in einer Übersicht die Themen des Kontaktunterrichts im Semester.

SW	Thema	Referenzen zur empfohlenen Literatur (s. Abschnitt 7)
01	Einführung und Überblick	[1] Kap. 1-3 [2] Kap. 1-2
02	Anforderungsanalyse I	
03	Anforderungsanalyse II	[1] Kap. 4-7, 10-12, 30 [2] Kap. 3
04	Domänenmodellierung	[1] Kap. 9, 31
05	Quiz 1: Analyse Softwarearchitektur und Design I	[1] Kap. 13, 33-35, 39 [3] Kap. 15-32
06	Softwarearchitektur und Design II	[1] Kap. 14-18, 23-25, 28-29 [2] Kap. 4-8 [3] Kap. 7-11
07	Use-Case Realisation	
08	Entwurf mit Design Patterns I	[1] Kap. 26
09	Entwurf mit Design Patterns II	[1] Kap. 36
10	Implementation, Refactoring und Testing	[1] Kap. 19-22
11	Quiz 2: Design Vertiefung 1: Gemäss Absprache aus 3.1	
12	Vertiefung 2: Gemäss Absprache aus 3.1	
13	Vertiefung 3: Gemäss Absprache aus 3.1	
14	Wrap-up der Vorlesung	

3.1 Vertiefungsthemen

Es stehen 4 Themen zur Verfügung. In Absprache mit den Studierenden werden maximal 3 Themen gezeigt.

- Client Server
- GUI Architekturen
- Persistenz
- Framework-Design



4 Unterrichtsbetrieb

Das Modul SWEN1 umfasst 4 Lektionen pro Woche. Diese 4 Lektionen sind in der Regel so aufgeteilt, dass in 2-3 Lektionen Theorie vermittelt wird und in 1-2 Lektionen Aufgaben zur Vorlesung bearbeitet werden (Modultyp mit integriertem Praktikumsteil).

5 Leistungsnachweise

Während des Semesters (30% des gesamten Leistungsnachweises):

Im laufenden Semester finden zwei Quizzes statt (jeweils 10 Multiple-Choice-Fragen, Zeitdauer 15'). Diese Quizzes dienen der formativen Lernkontrolle. Die erreichten Punkte ergeben eine Note, wobei 80% der erreichbaren Punkte die Note 6.0 ergeben. Diese Note trägt mit Gewichtung 0.1 zur Semesternote bei.

Im laufenden Semester finden verschiedene Präsentationen und Aufgaben im Bereich Lernaufgaben / Wissenssicherung / Inhalte zu den aktuellen Themen statt. Die Vorbereitung und Durchführung dieser Beiträge durch die Studierenden werden mit Punkten bewertet. Sie dienen der aktiven Aufarbeitung der Inhalte und der engeren Verbindung zu PM3.

Die Punkte pro Lerneinheit leitet sich aus den vergebenen Punkten ab.

- 0 Punkte (nicht teilgenommen oder nichts beigetragen)
- 1 Punkt (minimaler Beitrag)
- 2 Punkte (mit Einsatz einen substantziellen Beitrag geleistet)
- 3 Punkte (mit Einsatz einen perfekten Beitrag geleistet)

Die Summe der erreichten Punkte wird den Punkten der SEP hinzugezählt, wobei die Punkte im Verhältnis umgerechnet werden, so dass von den maximal erreichbaren Punkten 80% aus der SEP und 20% aus den Unterrichtsaufgaben stammen (Beispiel: SEP mit max. 72 Punkten, 10 bewertete Unterrichtsaufgaben mit max. 30 Punkten -> die erreichten Punkte der Unterrichtsaufgaben werden auf 18 Punkte heruntergerechnet).

SEP (70% des gesamten Leistungsnachweises):

Es findet eine obligatorische Semesterendprüfung (SEP) statt. Sie dauert 90 Minuten.

- Findet die SEP als Papierprüfung statt, sind als Hilfsmittel eine Zusammenfassung von 6 A4-Seiten (= 3 A4-Blätter vorne und hinten oder 6 A4-Blätter einseitig beschrieben/bedruckt), die Cheat Sheet UML und Design Patterns (s. Allgemeines Unterrichtsmaterial auf Moodle) zugelassen. Andere Unterlagen und elektronische Hilfsmittel wie Notebook, Handy etc. sind nicht zugelassen.
- Findet die SEP als elektronische Prüfung statt, so sind alle abgegebenen Unterlagen und die Inhalte der Praktika zugelassen (eine eigene Zusammenfassung wird dringend empfohlen).

Es gibt keine weiteren Testat-Bedingungen.

6 Unterlagen

Alles Unterrichtsmaterial ist im entsprechenden Kurs zu SWEN1 auf Moodle abgelegt.

7 Empfohlene Literatur

Hier ist empfohlene Lektüre als Ergänzung und Vertiefung zum abgegebenen Unterrichtsmaterial aufgeführt. Das Buch von Larman [1] wird insbesondere Studierenden empfohlen, die noch wenig Vorwissen in der Softwareentwicklung haben.

[1] Larman, C.: UML 2 und Patterns angewendet, mitp Professional, 2005



- [2] Seidel, M. et al.: UML @ Classroom: Eine Einführung in die objektorientierte Modellierung, dpunkt.verlag, 2012
- [3] Martin, R. C.: Clean Architecture: A Craftsman's Guide to Software Structure and Design, mitp Professional, 2018

Weitergehende Lektüre zur Vertiefung eines bestimmten Themas ist jeweils noch in der entsprechenden Präsentation aufgeführt.