Course PM

DIT184, Software Analysis and Design, 7,5 Hec, Ht, 2019

**Examiner:** Michel Chaudron

**Course responsible:** Dave Stikkolorum

**Course Assistants/supervisors:**

* Omid Manai
* Rashad Kamsheh
* Yazan Alsahhar
* Majd Al Hatoum
* Ameera Darwish

**Lecturer:**Dave Stikkolorum / Michel Chaudron

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Course content:**

The purpose of this course is to familiarize students with concepts, methods, and tools

for object-oriented analysis and design of software systems, with emphasis on methods

applied in large product development projects. The course introduces common design

principles and patterns that support the development of maintainable, reusable and

extensible software. The course gives an introduction to UML.

Analysis- and design models are expressed using UML models such as use case

diagrams, class diagram, sequence diagrams, and state diagrams. Furthermore,

techniques and guidelines are introduced for analysis of software domain and

requirements.

**Learning outcomes:**

On successful completion of the course the student will be able to:

*Knowledge and understanding*

* explain how to represent a software system using UML models,
* explain guidelines and heuristics for performing a domain analysis,

*Skills and abilities*

* analyze and design software systems using object-oriented techniques,
* create an UML model that is an abstract representation of the source code,
* use tools for domain and requirements analysis, modeling, program visualization, and object-oriented program design,

*Judgement and approach*

* analyze how software design principles and patterns impact software quality,
* reflect on - and resolve inconsistencies between various models used as part of a single system’s design.

**Course structure/course implementation:**

The course consists of a series of lecture sessions. The lectures are complemented by practical assignments done in pairs. The assignments are thematic, they put in practice the concepts and techniques discussed in the lectures. Assignment are take home and their respective solutions are discussed during supervision sessions. Further, the lectures are complemented by further readings.

**Examination forms:**The course is examined by an individual, written, hall examination (4.5 hec) and practical

assignments (3 hec) carried out in pairs.

**Course Literature:**Object-Oriented Systems Analysis And Design Using UML, by Simon Bennett, Ray Farmer and Steve McRobb ( 2010, Edition: 4 - ISBN: 9780077125363 )

Ambler, S. W. (2003). Agile model driven development is good enough. IEEE Software, 20(5), 71-73. (<http://ieeexplore.ieee.org/abstract/document/1231156/>)

Agile Modeling, An Overview, Scott W. Ambler (<http://www.agilemodeling.com/shared/AMPamphlet.doc>)

Martin, Robert C. "Design principles and design patterns." Object Mentor 1.34 (2000). (<http://staff.cs.utu.fi/staff/jouni.smed/doos_06/material/DesignPrinciplesAndPatterns.pdf>)

**Schedule:**

Lecture and supervision planning, including reading guide

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Lecture | Topics | Reading |
| March 26 | L1 | Introduction (course logistics, scoping, etc.)Design Process (Analysis & Design)Modeling in general | Book: Chapter 1 (1.1-1.5), 3 (only 3.3, 3.4), 2, 5 |
| March 28 | L2 | Systems/Responsibilities/Roles,Components, Objects (incl. OO orientation), Context diagram | Book: Chapter 1,2,4,5 |
| March 29 | S1 | Supervision assignment 1 | Canvas: Assignment 1 – Dutch tulips  |
| April 2 | TA | No lecture, TA session in lecture time slot |  |
| April 4 | L3 | The Problem Domain (Analysis): Stakeholders, use cases, use case Diagrams | Book: Chapter 6 |
| April 5 | S2 | Supervision assignment 2 | Canvas: Assignment 2 |
| April 9 | L4 | Domain Modelling with Class Diagrams (Analysis / Finding Objects) | Book: Chapter 7,8 (8.1-8.4) |
| April 11 | L5 | CRC cards | Book: Chapter 7.6 |
| April 12 | S3 | Supervision assignment 3 | Canvas: Assignment 3 |
| April 16 | L6 | Collaborations / Internal and external interaction with sequence diagrams / State Machine Diagrams / Activity Diagrams | Book: Chapter 5.3, 9 (except 9.5, 9.6), 10 |
| May 2 | L7 | State Machine Diagrams / Activity Diagrams | Book: Chapter 11 |
| May 3 | S4 | Supervision assignment 4 | Canvas: Assignment 4 |
| May 7 | L8 | Going from Domain to Technical Design (more depth/detail on UML notation), layeringComponent Diagram (part 1) - **Design Principles** | Book: Chapter 12, 13(13.1 – 13.5.2), 14Martin, Robert C. "Design principles and design patterns." Object Mentor 1.34 (2000). [[PDF] utu.fi](http://staff.cs.utu.fi/staff/jouni.smed/doos_06/material/DesignPrinciplesAndPatterns.pdf) |
| May 9  | X | No lecture | X |
| May 10 | S5 | Supervision assignment 5 | Canvas: Assignment 5 |
| May 14 | TA | TA session at lecture slot, Optional for lecture |  |
| May 16 | L9 | Going from Domain to Technical Design (more depth/detail on UML notation), layeringComponent Diagram (part 2) **Design Patterns** | Book: Chapter 14, 15, 19.3 |
| May 17 | L10 | Agile development with UMLAgile Software Development Workshop | Book: Chapter 2, 21<http://ieeexplore.ieee.org/abstract/document/1231156/><http://www.agilemodeling.com/shared/AMPamphlet.doc> |
| May 17 | S6 | Supervision assignment 6 | Canvas: Assignment 6 |
| May 21 | TA | TA session at lecture slot |  |
| May 23 | L11 | Recap Design PrinciplesSoftware Design Patterns | Book: Chapter 11, 12, 13, 14, 15Martin, Robert C. "Design principles and design patterns." Object Mentor 1.34 (2000). [[PDF] utu.fi](http://staff.cs.utu.fi/staff/jouni.smed/doos_06/material/DesignPrinciplesAndPatterns.pdf) |
| May 24 | X | No lecture |  |
| May 28 | X | No lecture |  |
| June 7 |  | Exam |  |