CT60A5103 Software Engineering Models and Modeling

The key information on this course has been summarized here from LUT SISU, https://sis-lut.funidata.fi/student/courseunit/lut-CT60A5103-2019-2020/brochure .

Teacher in Charge: Associate Professor Antti Knutas

Year: MSc (Tech.) 1

Course Schedule: Fall periods 1-2

Assessment scale and assessment methods:

Grade: 0-5, final project 50 %, Moodle online exam 50%. The course involves mandatory weekly exercises.

Aims:

Software modeling (this course) is aimed at reducing the gap between problem and software implementation through the development and use of models, which describe complex systems at multiple levels of abstraction and from a variety of perspectives. A model is an abstraction (one aspect or entire system) of an existing or planned system. Models are created to serve particular purposes, for example, to present a human-understandable description of some aspect of a system or to predict its quality.

The course is focused at building a deep understanding of the concept of model and modeling while enabling the students to be able to:

1. Master the importance of conceptual modeling techniques in software engineering and the diverse types of models.
2. Explain the concepts of meta-models, platforms dependent and independent models, model-to-model transformations, automated code generation from models.
3. Understand and select the appropriate modeling method or methods for the software development project at hand and for the various types of software systems such as critical-safety systems, interactive consumer services, enterprise applications, hardware software, etc.
4. Manage, plan, analyze and contribute to various models to represent requirements, design, implementation and maintenance of large intensive software products, systems and services.
5. Understand how human, social and technical factors may have (both) positive and negative influence on the methods and practices of modelling in software engineering.
6. Identify the modeling challenges facing the software engineering research community as well as the avenues for further investigations.

Contents:

Modeling in Software Engineering Body of Knowledge (SWEBOK). Principles and foundations of software engineering. Formal methods. Prototyping techniques. Object-oriented modeling. Data-centric models. Model-driven architecture (MDA). Modeling techniques. Importance of modeling in software development projects and processes.

Teaching Methods:

Online lectures 18 h, exercise classes 16 h, independent reading 18 h, weekly projects 32 h, 1st period. Final project 70h and preparing for the exam 8 h, 2nd period. Total 162 h.

Prerequisites:

Bachelor's thesis or degree completed.

Teaching Language:

English

Study materials:

The students are encouraged to walkthrough, one of the two following books as a basic introductory reading:

(1) R.S Pressman. Software Engineering: A Practitioner’s Approach, 7/e, McGraw Hill, 2010

(2) J. Sommerville. Software Engineering. 9/e, Addison Wesley, 2011.