Jnr: 2018/2845-3.1.2.2

# Course syllabus

## Faculty of Technology

Department of Computer Science and Media Technology

## 4DV651 Projekt i modell-baserad utveckling, 10 högskolepoäng
## Project in Model-based development, 10 credits

**Main field of study**
Computer Science

**Subject Group**
Informatics/Computer and Systems Sciences

**Level of classification**
Second Level

**Progression**
A1N

**Date of Ratification**
Approved by Faculty of Technology 2018-10-08
The course syllabus is valid from autumn semester 2019

**Prerequisites**
90 credits in Computer Science (including a degree project at Bachelor level).

## Objectives

Upon completion of the course, the student should be able to:

- Classify and explain key principles and concepts within model-based development, such as models, meta-models, constraints, transformations, semantics, and abstract and concrete syntax, and
- describe the architecture of current modeling frameworks as well as how domain specific modeling frameworks can be formulated using these.
- Independently learn how to use different frameworks and tools for model-based development,
- implement a system that fulfills a set of requirements that were elicited from a customer,
- given a set of models and framework, create model editors, model checks, and model transformations,
- use model-based development to implement an executable software program and verify its properties, and
- plan an agile project, including requirements elicitation, prioritization, and time estimation.
- Critically reflect on the benefits and drawbacks of model-based development from a software engineering perspective, e.g., the time different tasks takes, how easy the tools are to use, etc,
- reason about the benefits and drawbacks of model-based development from a societal perspective, e.g., with respect to security and safety requirements, economic benefits, etc., and

- analyze how well the agile way of working worked within a project from a working environment perspective and suggest improvements.

## Content

The course is a project course that, given a realistic problem and settings, considers the entire CDIO (Conceive-Design-Implement-Operate) cycle. The students are placed in the role of a development team within an agile organization with many requirements on their software.
The students are expected to work using agile processes in teams of 5-7 and are expected to perform all roles except product owner. Special focus is given to requirements elicitation, description, planning, and estimation from an agile perspective as well as documentation.

The following topics are covered:

- Advanced models and software development.
- Model-based development and architecture.
- Software problems, such as security, performance, and stability.
- Software for different domains, and the difference in requirements.
- The benefits and drawbacks of model-driven development.
- Modeling languages, meta-modeling, and profiles.
- Model transformations and model constraints.
- Action languages.
- Domain specific languages.
- Model-based testing.
- Model validation.
- Code generation from models.
- Tools for model-based development.
- Methods to estimate time in agile projects.
- Methods to elicit and document requirements
- Methods to document software in agile projects and the relationship between documentation, models, and source code.
- How to write reflection reports and post-mortem analyses of completed projects.

## Type of Instruction

The instruction consists of lectures and supervision meetings. The lectures present the project as well as the tools, methods, and resources that the students are expected to use. The students will have regular supervision meetings with a teacher during the project.

All projects will be presented at a seminar at the end of the course.

## Examination

The course is assessed with the grades A, B, C, D, E, Fx or F.
The grade A constitutes the highest grade on the scale and the remaining grades follow in descending order where the grade E is the lowest grade on the scale that will result in a pass. The grade F means that the student's performance is assessed as fail (i.e. received the grade F).

Assessment of student performance is made through vision and planning documents, project work (incl. deliverables), two reflection reports, and a presentation. Students who do not pass the regular examination will be offered retrials close to the regular examination.

To pass the course, grade E or higher is required for all parts. The final grade is decided from: vision and planning documents (20%), project work (50%), reflection report on model-based development (10%), reflection report on the project (10%), and final presentation (10%).

## Course Evaluation

During the course or in close connection to the course, a course evaluation is to be carried out. The result and analysis of the course evaluation are to be communicated to the students who have taken the course and to the students who are to participate in the course the next time it is offered. The course evaluation is carried out anonymously. The compiled report will be filed.

## Other

Grade criteria for the A–F scale are communicated to the student through a special document. The student is to be informed about the grade criteria for the course by the start of the course at the latest.

The course is conducted in such a way that the course participants' experiences and knowledge are made visible and developed. This means, for example, that we have an inclusive approach and strive for no one to feel excluded. This can be expressed in different ways in a course, for example by using the gender neutral example.

## Required Reading and Additional Study Material

The students are expected to find suitable literature on their own.

The list below can be used as a starting point.
- Brambilla, M., Cabot, J. and Wimmer, M., *Model-Driven Software Engineering in Practice*, second edition, Morgan & Claypool Publishers. 2017.

- Steinberg, D., Budinsky, F., Paternostro, M. and Merks, E., *EMF: Eclipse Modeling Framework*, second edition, Addison-Wesley Professional. 2008.

- Kelly, S. and Tolvanen, J-P., *Domain-Specific Modeling*, Wiley-IEEE Computer Society Press, 2008.

- Mellor, S. J. and Balcer, M. J., *Executable UML: A Foundation for Model-Driven Architecture*, Addison-Wesley Professional. 2002.

- Royer, J. and Arboleda, H., *Model-Driven and Software Product Line Engineering*, John Wiley & Sons, Inc. 2013.