
LUT
University

Software Engineering Models and Modeling

Course introduction and arrangements

Antti Knutas

Course introduction

Who and what?

Course staff

- Associate Professor Antti Knutas
- Software engineer, researcher and educator (practitioner in the past)
- Dozens of arranged course instances in three universities, including software engineering models and modelling, sociotechnical systems, research mentoring, code camps, and programming (front, web, databases, etc.)

Course TAs, very talented software engineering practitioner and researcher, MSc. Tanvir Hasan

From the study guide, course content

- Master the importance of conceptual modelling techniques and diverse types of models
- Understand and select appropriate modelling methods
- Manage, plan, analyse and contribute to various models to represent requirements, design, implementation, and maintenance
- Understand how human, social, and technical factors affect these models
- Identify upcoming modelling challenges facing software engineering research community

A model is an abstraction (one aspect or entire system) of an existing or planned system.

Software modelling 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.

What do we really study (study guide translated)

- To first understand and then communicate to others how complex software systems work.
- Use visualization as a step of problem solving between problem definition and software implementation
- Turn *knowledge* about these issues into *skills*

We'll also do a) a quick recap of software engineering fundamentals today and b) connect the field of software modelling to software engineering
(motivation: we have diverse group of MSc. participants – some of you already are, the rest will *become* software engineers during this degree)

Disclaimer: You will most likely need *some* of the tools of this course in your work. You will rarely need *all* modelling tools in each project.

Prerequisites vs. content?

We have a large and a diverse set of participants: People new to LUT, people continuing from our bachelor's, or people taking this course as a minor.

Some content might be a recap. Some content might be completely new. Some people might need to do some background reading on software engineering before studying course materials.

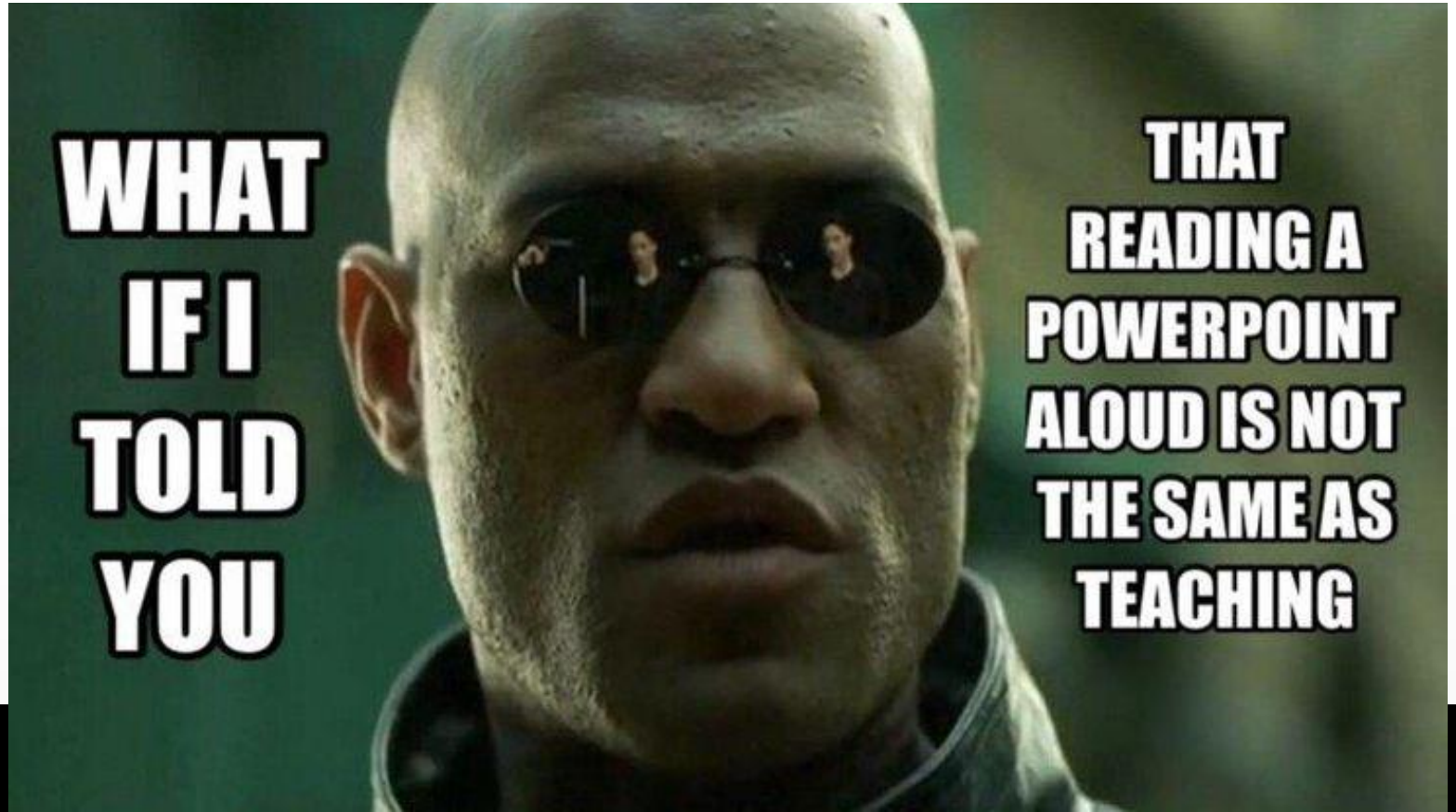
My advice: Take what you find useful and go with it. If the material is already familiar, then check your knowledge with the assignments or exercises.

If you are not coming from a software engineering bachelor's (or have not taken our 1st BSc. year software engineering course), I recommend reading additional sections from Sommerville's coursebook and not just the recommended reading parts.

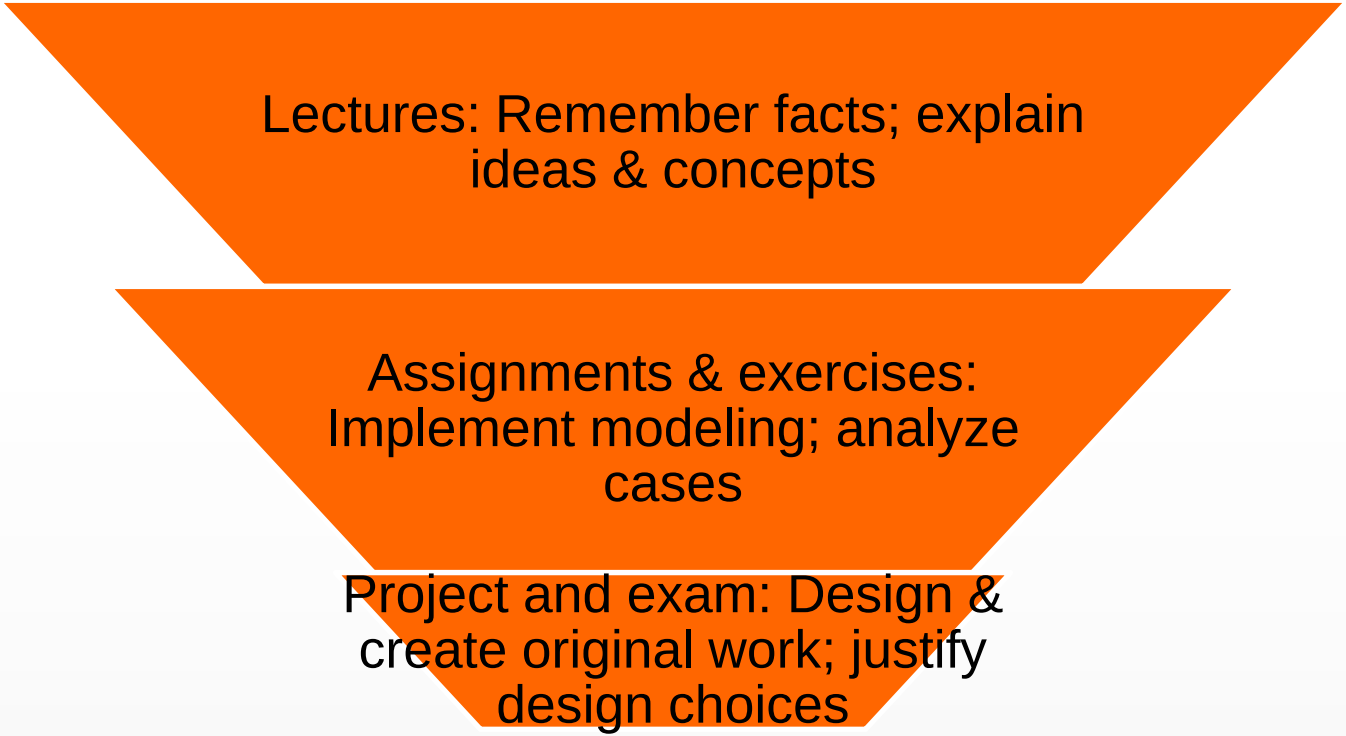
Basic principles

How do we work?

Teaching (learning) principles: Active learning and constructive alignment



Course components, visually (roughly following Bloom's taxonomy)



- Seminars: A small bonus (TBD)
- Assignments: +8% to 16% bonus
- Project: 100% of grade (optional exam)

How do we interact?

- Lectures mainly async – providing theoretical background in advance to practical work
- Interactive discussion events (discussion-based lectures at LPR and Lahti seminars)
- Exercises during 1st period and mentoring meetings for project teams during 2nd period
- Through your submitted assignments and project work – these are the basis for your grade (demonstration of your skills through practical work)

Flipped classroom and constructive alignment

- Flipped classroom, as applied here: Theory is learned independently, either through attending lectures, reading books, or watching video lectures
  - Materials are provided in advance and then prepared but possibly inexperienced people interact together
  - Maximal learning benefits come through working together on assignments and discussing challenges with course staff or classmates during exercises
    => learning to apply the skills in practice
- *Design is a skill, not a knowledge*
- Constructive alignment (principles)
  - Learning goals should be clear, serve a purpose, and set in advance.
  - Students need to be placed in situations and environments that elicit the required learnings, with declarative teaching minimized.
  - Students are then required to provide evidence, either by self-set or teacher-set tasks, as appropriate, that their learning can match the stated objectives.

Teamwork?

You may return all assignments alone. However, we'll also allow returning all assignments and the course project in a team of three people.

Please come to the first exercises either live or in Zoom (next week) and try to chat and form teams!

After meeting each other, you may of course start using any teamwork, chat or collaboration software you'd like.

(extra credit tasks and course exam is the only activity to be done alone)

Regarding academic dishonesty

This is not a programming course; but programming knowledge is required. The course project and assignments might require actual programming work.

This course is not difficult; anyone with BSc. level software engineering knowledge, willingness to read all the materials, and patience can pass the course. However, this course will involve time, work, and patience.

Discussing with your peers is allowed and you can recommend them good reading sources or videos; sharing solutions is not. Do not tempt your friends by showing them the finalized solution. Do not copy or buy your project work, either. Your return materials are automatically checked against each other and the Internet.

Do take care, be proud of your work, and present original materials. Even if you are participating online, this is no longer a MOOC. If ever in doubt about LUT academic integrity guidelines, do contact the course staff.

Lastly, take a very careful look at our generative AI policy. In short, you may use any tools at your disposal, but you must document the prompts and be transparent about which tools you used, use your own critical thinking and take ownership of the final product.

Course content

What do we do?

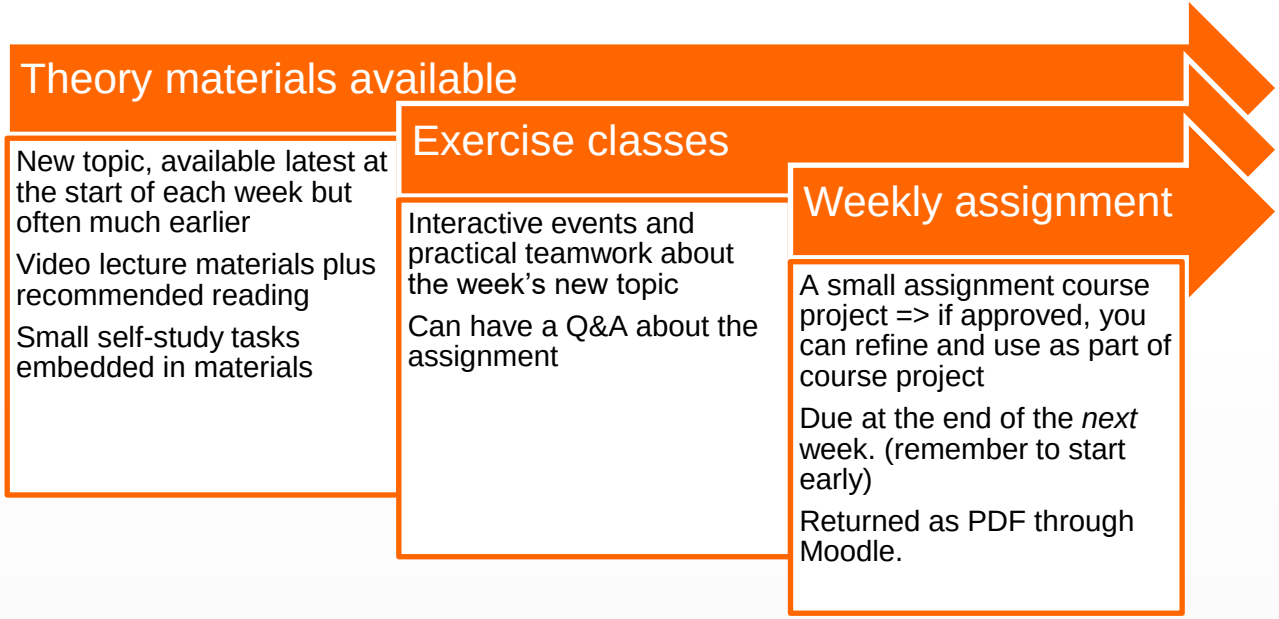
Course structure

- 6 ECTS (converts to approximately 156h of work)
  - => Online lectures 16 h, exercise classes 16 h, independent reading 16 h, weekly projects 30 h, 1st period. Final project 70h and preparing for the exam 8 h, 2nd period. Total 156 h.
- Online lectures and exercises to learn the concepts, tools, and practises (1st period)
- Project work where these are applied to a practical problem (2nd period)
- Optional ”EXAM” classroom exam

A rough lesson plan for period 1 (subject to change)

1. Course introduction and software engineering fundamentals
2. Process models.
3. Modelling software with objects.
4. Business and domain models. Intro to requirements.
5. From requirements to models. MVC/BCE architecture models.
6. Architectural modeling.
7. Software process models in the industry. Modeling critical systems and risks.
8. Constraints of modeling in software engineering. Introducing course project.

Weekly timeline for period 1



Weekly assignments

These generally involve looking at the large course project and completing one part of it for a pass/fail evaluation. If passed, you may reuse (copy-paste and improve) that part for the course project.

If possible, team up as early as possible and work on the assignments. This way finalizing the course project is easy, since you can use all of the small, pre-existing pieces.

Lastly, final project assessment is more strict than pass/fail and expects more, so please do remember to revise and improve.

Period 2?

A large, independent design and implementation project. Course activities will involve mentoring and feedback meetings.

Key course materials, assignments, and activities

- Materials
  - Our main coursebook: Ian Sommerville. 2010. Software Engineering (any ed.). Addison-Wesley Publishing Company, USA.
  - Other eBooks, as linked from the lectures tab (once lecture is published).
  - Asynchronous lectures & self-study activities in Moodle (incl. reading recommendations)
- Tasks (assignments, exercises, project, exam)
  - Weekly learning assignments: Design, essays, or practical tasks
  - A final project during period 2
  - Exam at the end of period 2 (EXAM, self-scheduled)
- Activities
  - Exercise classes in Lappeenranta, Lahti and online – ask anything, see demos, and work together in an interactive environment with Tanvir
    - => all exercise classes have the same content (select the one whose location and time suits you best)
  - Mentoring meetings in period 2

Course grading

- Course project: 100 pts
- Optional ”EXAM” exam: +40 pts (you may revise two sections of the project if some part did not go well)
- Assignments: max 10 bonus points; you must return a minimum of one successfully to pass the course

Plus a selection of small bonus points from participation or activity.

Grade: 0-5; calculated from points as follows: 0-49=0; 50-59=1; 60-69=2; 70-79=3; 80-89=4; 90 and beyond = 5.

Course grading and requirements, more detail

Activity	Min. requirements	Max. points
Course project	50 pts.	100 pts.
Exam	0 pts.	approximately +40 pts. (if the project groupwork does not go well, you may take the optional exam, select two sections of the project and then redo them by yourself)
Weekly assignment	One assignment (2 pts.)	+ 10 pts.
Exercise participation and other activity bonus	0 pts.	+8 pts. from participation OR +8 pts from Lahti seminars
Answering the end of course feedback survey	0 pts.	1 pts.

Why a project AND an exam?

There is an optional chance to revise any part of the project

=> Exam is optional if you are happy with the course project

How to get started with and how to complete the course?

What we recommend for you to do in order to get started (and what is the minimum to pass)

How to start?

- If you are spending 100% of time in degree program: attend weekly exercises if you can, meet students, form up a group and start working on course project week by week (watch lectures together; read coursebook + tutorials as needed)
- If you are working and studying: watch lectures, attend online exercises if you can, team up or work solo; start working on course project early
- Review and complete weekly assignments right away. They become part of the course project.

There is a lot of flexibility: most learning opportunities are optional and you choose what to attend (assessment at end of course)

What is needed to complete the course? Only the course project.

You will get assessed based on skills demonstrated at the end of the course => a lot of responsibility. (You will get bonus points from weekly assignments and other minor tasks.) Weekly assignments almost directly become building blocks of the course project.

We are here to support you, but there is a lot of flexibility. You can attend interactive events or work on your own with the support of recordings, online tutorials, and independently with your team.

Having said that, I *hope* you will come meet other upcoming professionals and work with them. In addition, if you do choose to come to events, we are here to work with you and excited to share what we know on the topic.

Course tools

What facilitates our work?

Course tools

- Moodle: All materials, assignments, news, links
  - All assignments are given in writing in Moodle and returned to Moodle, according to instructions – emailed returns cannot be included in grading
  - New reading recommendations each week in period 1 (which book chapters to cover)
- Zoom – synchronous video lectures where suitable, online exercises, and online office hours

Let’s have a look at course Moodle

⇒ Where to find materials

⇒ Where to return assignments

⇒ Other features

Reaching out to course staff

- Consider using the public chat or forums first – this way the solution is shared to all of the learners
- If it is private or not general, reach out to staff by email
  - Antti Knutas (antti.knutas@lut.fi) on grading structure, exceptions to deadlines (illness etc.), and general course issues
  - Tanvir Hasan (Md.Tanvir.Hasan@lut.fi) on individual grading decisions, help for assignments, locating materials
- It’s better to reach out to anyone than not reach out, though – we’re friendly!

Concluding with a Q&A