



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DIT257 Agile software project management, 7.5 credits

Agil projektledning, 7,5 högskolepoäng

First Cycle

Confirmation

This course syllabus was confirmed by Department of Computer Science and Engineering on 2019-11-18 and was last revised on 2022-11-23 to be valid from 2023-08-28, autumn semester of 2023.

Field of education: Science 100%

Department: Department of Computer Science and Engineering

Position in the educational system

The course is offered within the framework of several degree programmes. The course is also a single subject course at the University of Gothenburg. The course is part of the following programmes: 1) Computer Science, Master's Programme and 2) Computer Science, Bachelor's Programme

The course can be part of the following programmes: 1) Computer Science, Master's Programme (N2COS), 2) Applied Data Science Master's Programme (N2ADS) and 3) Computer Science, Bachelor's Programme (N1COS)

Main field of studies

Computer Science

Specialization

G2F, First cycle, has at least 60 credits in first-cycle course/s as entry requirements

Entry requirements

To be eligible for the course, the student must have completed courses in programming for at least 15 HEC, e.g., two of DIT440, DIT012, or DIT953.

Learning outcomes

After completion of the course the student is expected to be able to:

Knowledge and understanding

- describe software engineering as an engineering discipline by using relevant terminology
- describe the relationship between stakeholder, product, and process

Competence and skills

- specify, implement, and evaluate a system based on what different stakeholders perceive as valuable
- learn tools and APIs which are relevant for the project in collaboration with the other team members
- apply a structured software development process as a member of a team

Judgement and approach

- reflect on how the software development process was applied in a project
- reflect on your own and the team's learning strategies

Course content

The course provides a practical introduction to Software Engineering. Students work on an open problem that is defined by stakeholders outside of the students' team. This means that students will not be able to define the project they work on themselves. By iteratively planning and evaluating their work, they organise themselves in order to reach a joint goal with limited resources. To do this, they apply an agile software development process to structure their work. They specify and evaluate requirements and the collaboration with stakeholders to assure that what is being delivered is perceived as valuable. Students reflect on their own work and learning every week to enable a continuous improvement of their way of working. At the same time, they acquire skills and knowledge in new technologies, tools, and fitting ways to use them in order to provide value to a stakeholder based on the students' own learning strategies.

*Sub-courses***1. Project (Projekt), 7.5 credits**

Grading scale: Pass (G) and Fail (U)

Form of teaching

The course is organized as a project where the students work in teams of usually six students to address a real-world software engineering task. The teams have weekly supervision meetings. The project is supplemented by lectures that provide insight into the task the students are working on and software engineering in general.

Language of instruction: English

Assessment

The course is assessed continuously through a weekly report. The report contains both a joint team part and an individual part for each team member which documents the project's progress. The final grade will be based on the last version of the report according to the grading criteria below. Students thus iteratively and incrementally add to their reflections throughout the project. The content and structure of the report is aligned with the intended learning outcomes. The teachers supply detailed questions and topics to reflect on for each course instance.

If a student, who has failed the same examined component twice, wishes to change examiner before the next examination, a written application shall be sent to the department responsible for the course and shall be granted unless there are special reasons to the contrary (Chapter 6, Section 22 of Higher Education Ordinance).

In cases where a course has been discontinued or has undergone major changes, the student shall normally be guaranteed at least three examination occasions (including the ordinary examination) during a period of at least one year from the last time the course was given.

Grades

The grading scale comprises: Pass (G) and Fail (U).

The grading scale comprises: Pass (G) and Fail (U).

An individual passes the course when both the team part and the individual part are passed.

The team part consists of a final report that summarises and synthesises the weekly reflections. The reflections should clearly detail the teams' own experiences, the proposed changes to improve on the topic, how the changes were evaluated (KPIs, acceptance testing etc.) and subsequent experiences and changes. In addition, the team needs to submit the source code, tests, a breakdown of contributions, the technical documentation, instructions on how to install and use the product, an executable if applicable (such as an apk- or jar-file) as well as the weekly reflections.

The individual part consists of weekly reflections, source code contributions, and a peer assessment. Complete weekly individual reflections have to be submitted. Students need to reflect on their own learning and how the individual themselves contributed to the process and to the team. There should be a record that shows that an individual has also contributed to the source code and the peer assessment should indicate that the individual participated in a meaningful way.

Course evaluation

The course is evaluated through meetings both during and after the course between teachers and student representatives. Further, an anonymous questionnaire is used to ensure written information. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed. The results of and possible changes to the course will be shared with students who participated in the evaluation and students who are starting the course.

Additional information

The course is a joint course together with Chalmers.

Course literature will be announced at the latest 8 weeks prior to the start of the course.

The course replaces the course DIT543 Software engineering project, 7.5 credits. The course cannot be included in a degree which contains DIT543. Neither can the course be included in a degree which is based on another degree in which the course DIT543 is included