



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DIT593 Software engineering principles for complex systems, 7.5 credits

Programvaruteknik för komplexa system, 7,5 högskolepoäng

First Cycle

Confirmation

This course syllabus was confirmed by Department of Computer Science and Engineering on 2020-10-27 to be valid from 2021-08-30, autumn semester of 2021.

Field of education: Science 100%

Department: Department of Computer Science and Engineering

Position in the educational system

The course can be part of the following programmes: 1) Computer Science, Bachelor's Programme (N1COS) and 2) Software Engineering and Management Master's Programme (N2SOF)

Main field of studies

Software Engineering

Computer Science-Software Engineering and Tech

Specialization

G1F, First cycle, has less than 60 credits in first-cycle course/s as entry requirements

G1F, First cycle, has less than 60 credits in first-cycle course/s as entry requirements

Entry requirements

The student must know Java programming and must have successfully completed courses on data structures and algorithms (e.g., DIT181) as well as on object-oriented programming (e.g., DIT042, DIT953).

Applicants must prove knowledge of English: English 6/English B or the equivalent level of an internationally recognized test, for example TOEFL, IELTS.

Learning outcomes

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

Identify and reason about recurrent problems of engineering complex systems and being able to apply appropriate solutions. The learning is driven by a concrete example of a software engineering or re-engineering project that will be developed in group work.

Knowledge and understanding

- Explain the challenges of engineering complex software systems
- Explain industrial practice and examples of complex software systems engineering
- Explain processes and concepts for engineering complex and variant-rich software systems
- Explain business-, architecture-, process-, and organization-related aspects of engineering complex software systems

Competence and skills

- Model a software system from different perspectives (e.g., using feature models, UML diagrams, architecture description languages)
- Engineer a variant-rich software system (e.g., variant-rich software system, software product line, software ecosystem)
- Analyze and re-engineer a complex software system -Use and reason about modularization techniques
- Use modern component or service frameworks

Judgement and approach

- Analyze existing software systems and discuss potentials for improvement or re-engineering
- Reason about characteristics software modularity concepts
- Recognize in which situations which principles for handling of complex software systems are appropriate
- Read and analyze scientific literature

Course content

Programming expertise is only one of many skills required to engineer complex software systems. In this course we will critically analyse what software-engineering principles support the engineering of complex software systems. We will discuss these principles in the lectures and will apply them in project work.

Sub-courses

1. Assignments (*Inlämningsuppgifter*), 1.5 credits

Grading scale: Pass with distinction (5), Pass with credit (4), Pass (3) and Fail (U)

2. **Project** (*Projekt*), 6 credits

Grading scale: Pass with distinction (5), Pass with credit (4), Pass (3) and Fail (U)

Form of teaching

There will be weekly lectures covering the theoretical course content. Additionally, there will be project work in groups and, as a part of this, weekly compulsory supervision meetings in the groups. In addition, there will be individual written assignments.

Language of instruction: English

Assessment

The examination consists of two parts: an individual written report assignment (1.5 hec) and a group project (6 hec). The project part is graded individually, taking into account the group work as well as the student's individual contribution to the groupwork.

If a student, who has failed the same examined element on two occasions, wishes to change examiner before the next examination session, such a request is to be submitted to the department in writing and granted unless there are special reasons to the contrary (Chapter 6, Section 22 of Higher Education Ordinance).

In the event that a course has ceased or undergone major changes, students are to be guaranteed at least three examination sessions (including the ordinary examination session) over a period of at least one year, though at most two years after the course has ceased/been changed. The same applies to work experience and VFU, although this is restricted to just one additional examination session.

Grades

The grading scale comprises: Pass with distinction (5), Pass with credit (4), Pass (3) and Fail (U).

To pass the course, all mandatory components must be passed. To earn a higher grade than Pass, a higher weighted average from the grades of the components is required.

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.

Courses on user-interface development (e.g., TIG095), and on testing (e.g., DIT083) are recommended.

The course replaces the course DIT594, 7.5 credits. The course cannot be included in a degree which contains DIT594. Neither can the course be included in a degree which is based on another degree in which the course DIT594 is included.