



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DIT598 Software Evolution Project, 15 credits

Mjukvaruevolutionsprojekt, 15 högskolepoäng

Second Cycle

Confirmation

This course syllabus was confirmed by Department of Computer Science and Engineering on 2017-02-10 and was last revised on 2019-12-09 to be valid from 2020-08-31, autumn semester of 2020.

Field of education: Science 100%

Department: Department of Computer Science and Engineering

Position in the educational system

The course is a compulsory course in the Software Engineering and Management Master's Programme and an single subject course at the University of Gothenburg.

The course can be part of the following programmes: 1) Computer Science, Master's Programme (N2COS), 2) Software Engineering Master's Programme (N2SOM) and 3) Software Engineering and Management Master's Programme (N2SOF)

Main field of studies

Software Engineering

Specialization

A1F, Second cycle, has second-cycle course/s as entry requirements

Entry requirements

To be eligible for the course the student needs to have successfully completed the following courses, or equivalent:

- a 7.5 hec course in programming (e.g., DIT948 Programming),
- a 7,5 hec software project course (e.g., DIT191 Agile development processes),
- a 7.5 hec course in software quality (e.g., DIT847 Software Quality),

and at least one of the following courses, or equivalent:

- 7,5 hec DIT284 Requirements Engineering,

- 7,5 hec DIT544 Advanced software architecture,
- 7,5 hec DIT596 Software Language Engineering for Domain Specific Languages, or
- 7,5 hec DIT848 Model based testing.

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:

Knowledge and understanding

- explain the notion of software evolution scenarios as defined by the Architecture Driven Modernization (ADM) task force of the Object Management Group (OMG),
- summarize state of the art in methods and tools for software evolution tasks, such as program comprehension and software refactoring,
- discuss the challenges associated with software evolution,
- explain current research trends in program comprehension, clone detection, and refactoring.

Competence and skills

- extract a software product's architecture from a given code base and evaluate the quality of the software product,
- implement one software evolution scenario as defined by the OMG, such as platform or language migration,
- implement changes to a software product that lead to an improvement of the product's quality (quality improvement task),
- make use of synergies between different improvements goals for the same product,

Judgement and approach

- detect and judge needs for quality improvement or evolution in an authentic software product,
- plan the use of appropriate methods and techniques for performing a software evolution scenario and a quality improvement task,
- judge needs for improvement of methods and tools to support software evolution,
- plan and evaluate ideas for new or improved tools.

Course content

The purpose of this course is to familiarise the student with situations, activities, and techniques typical for software evolution, such as adding a new feature, refactoring, automating variability or testing, improving performance, re-balancing non-functional

requirements, and so on. Furthermore, students learn to plan, implement, and reflect on software evolution scenarios and improvements.

Form of teaching

The course contains project- and problembased teaching. The course consist of a series of group and individual milestones, and supervision meetings during the semester. Groups and teachers agree on a software evolution scenario to be performed as a project during the course. In additon, lectures and workshops are used to provide the student with methods and knowledge about software evolution.

Language of instruction: English

Assessment

The course consists of one module which consists of four components.

The course is examined by the following four mandatory components:

- weekly group milestones that are submitted during the course,
- a written group report to be submitted at the end of the course,
- a documentation of the individual contribution to the final outcome of the group work,
- an individual written final report.

The group work is carried out in groups of normally 6-8 students. The student collects points for all four components. Furthermore individual optional milestones can be used to collect bonus points during the course.

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 with Distinction (VG), Pass (G) and Fail (U).

In order to be awarded the grade Pass (G) for the full course, the student needs to pass the four mandatory components: group milestones, group report, individual contribution and individual report. Furthermore, the student needs to collect 50% of the maximum points. In order to be awarded the grade Pass with Distinction (VG) for the

full course, the student needs to collect 80% of the maximum points.

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.

Additional information

The course is a joint course together with Chalmers.

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

The course replaces the DIT599 Software Evolution Project 15 hec course. The course cannot be included in a degree which contains DIT599. Neither can the course be included in a degree which is based on another degree in which the course DIT599 is included.