



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### **DIT847 Software Quality, 7.5 credits**

Mjukvarukvalitet, 7,5 högskolepoäng

*Second Cycle*

---

#### **Confirmation**

This course syllabus was confirmed by Department of Computer Science and Engineering on 2014-10-27 and was last revised on 2019-02-07 to be valid from 2019-09-02, autumn semester of 2019.

*Field of education:* Science 100%

*Department:* Department of Computer Science and Engineering

#### **Position in the educational system**

The course is compulsory within Software Engineering and Management Master's Programme and is also a 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) Game Design & Technology Master's Programme (N2GDT), 3) Software Engineering Master's Programme (N2SOM) and 4) Software Engineering and Management Master's Programme (N2SOF)

*Main field of studies*

Software Engineering

*Specialization*

A1N, Second cycle, has only first-cycle course/s as entry requirements

#### **Entry requirements**

To be eligible for the course, the student should have a bachelor's degree in Software Engineering, Computer Science, Computer Engineering, Information Technology, Information Systems, or equivalent, including a completed course in programming (e.g. DIT042 Object-oriented Programming, DIT012 Imperative Programming with Basic Object-orientation, DIT143 Functional Programming or equivalent), and a completed course in practical software development or software engineering project (e.g. DIT212 Object-oriented programming project, or DIT543 Software Engineering Project).

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**

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

#### *Knowledge and understanding*

- explain fundamental concepts in software quality (e.g., internal / external quality, as well as quality in use)
- describe and explain definitions and activities related to software testing, such as faults, failures levels of testing and test automation
- explain the concept of continuous integration and relate them to software development processes
- explain and discuss the importance of using statistical analysis methods to support decision related to software quality
- describe how sustainability can be seen as a quality attribute in software products

#### *Competence and skills*

- construct statistical models to analyse quality-related data from software development organizations, (e.g., different code review practices, quality indicators)
- construct quality assurance plans
- collect data to quantify and statistically analyse the quality of software products (e.g. based on the existing open source products)
- construct automated measurement systems for measuring quality of software products based on the data from the modern development tools
- use modern tools for visualization of trends in software quality

#### *Judgement and approach*

- assess the importance of software quality in relation to time and costs in modern software development projects
- relate software quality to societal aspects of software development
- assess the risks of distinct quality assurance processes in modern software development companies (e.g., the impact of choosing among different testing techniques)

### **Course content**

This course is aimed at students learn modern methods of quality assurance in software development, such as development for embedded systems and web applications.

The course introduces the students to the modern methods for quality assurance in software development, as well as methods for statistical analysis to support decisions related to software quality. The content of the course consists of lectures about relevant standards (e.g. ISO/IEC 9126, ISO/IEC 9000, ISO/IEC 25000) for quality assurance and standards for measurement of quality (e.g. ISO/IEC 15939).

The course prepares the students to work as quality engineers and quality managers in software development organizations.

#### *Sub-courses*

**1. Report (*Rapport*), 3 credits**

Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)

The sub-course is a report based on statistical analysis to evaluate trade-offs between different techniques in software quality. The purpose of the sub-course is to assess students practical skills taught during the course.

**2. Written exam (*Skriftlig salstenta*), 4.5 credits**

Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)

The sub-course is an individual, written hall exam with both theoretical and practical questions. The purpose of the questions is to assess the students knowledge of definitions and relevant research within the field.

#### **Form of teaching**

The course consists of lectures where theory is explained and by practical assignments that focus on the use of statistical analysis to support decisions related to software quality.

*Language of instruction:* English

#### **Assessment**

The student is examined by an individual written report and a written exam done individually in an examination hall.

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).

A Pass grade (G) for the entire course requires a Pass grade for all sub-courses. A Pass with Distinction grade (VG) for the entire course requires a Pass with Distinction (VG) on both sub-courses.

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