



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DIT843 Quality Assurance and Testing, 7.5 credits

Mjukvarukvalitetsäkring och testning, 7,5 högskolepoäng

Second 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 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) and 3) 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.

In addition, the student should have completed courses in:

- Programming (e.g. DIT042 Object-oriented Programming, DIT012 Imperative Programming with Basic Object-orientation, DIT143 Functional Programming or equivalent) and
- 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) and quality assurance models
- Describe how sustainability can be seen as a quality attribute in software products
- Describe and explain definitions and activities related to software testing, such as faults, failures levels of testing and test automation
- Explain current research trends within software quality and testing (e.g., test flakiness, search-based software testing)
- Explain the distinction between software verification and software validation
- Name and describe different testing techniques and approaches, such as mutation testing, automated test prioritisation, test case generation, exploratory testing, etc
- Explain the concept of continuous integration and relate them to software development processes

Competence and skills

- Define metrics required to monitor and analyse different quality attributes of software products
- Measure quality of software products based on the data from development environments and tools
- Explain and implement test activities in connection to different levels of testing (e.g., system, integration or unit level tests)
- Define and implement a test plan or a test scope based on quality goals
- Design appropriate and meaningful tests
- Apply automated testing technologies on realistic examples using relevant tools (e.g., automated regression testing on continuous integration pipelines)
- Use automated measurement systems for measuring quality of software products based on the data from the modern development tools
- Use modern tools for visualisation of trends in software quality

Judgement and approach

- Assess the importance of software quality in relation to time and costs in software development projects
- Assess the quality of sets of tests using software artefacts (e.g., execution logs, test scripts, history of test executions)

- Relate software quality to societal aspects of software development

Course content

The course addresses the following content:

- Definitions of software quality assurance, quality attributes, software metrics and models in quality management, internal quality and external quality;
- Definitions and differences between software verification and validation approaches;
- Definitions and practical application of software testing techniques and approaches related to test planning, design, execution and analysis.

The course prepares the students to work as quality or test engineers (or managers) in software development organizations.

Sub-courses

1. Report (*Rapport*), 3 credits

Grading scale: Pass with distinction (5), Pass with credit (4), Pass (3) 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 (5), Pass with credit (4), Pass (3) 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 practical assignments that focus on the use of testing tools and approaches to enable cost-effective testing.

Language of instruction: English

Assessment

The course is examined by an written report and an individual written examination carried out in an examination hall. The report is graded individually.

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 (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 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.

The course replaces the course DIT847, 7.5 credits. The course cannot be included in a degree which contains DIT847. Neither can the course be included in a degree which is based on another degree in which the course DIT847 is included.