DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DIT192 Agile Development Processes, 7.5 credits

Agila utvecklingsprocesser, 7,5 högskolepoäng

Second Cycle

Confirmation

This course syllabus was confirmed by Department of Computer Science and Engineering on 2019-04-15 and was last revised on 2019-12-16 to be valid from 2021-01-18, spring 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 the Game Design & Technology Master's Programme and is offered within several programmes. It 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) Applied Data Science Master's Programme (N2ADS), 3) Game Design & Technology Master's Programme (N2GDT) and 4) Software Engineering and Management Master's Programme (N2SOF)

Main field of studies	*Specialization*
Interaction Design	AXX, Second cycle, in-depth level of the course cannot be classified
Computer Science	AXX, Second cycle, in-depth level of the course cannot be classified
Software Engineering	AXX, Second cycle, in-depth level of the course cannot be classified

Entry requirements

To be eligible for this course the student should have

- a bachelor degree
- a successfully completed course in programming (e.g., DIT042 Object-oriented Programming, DIT012 Imperative Programming with Basic Object-orientation, DIT143 Functional Programming, or equivalent);
- a successfully completed project course (or bachelor thesis) in applied software development or software engineering (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

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

Knowledge and understanding

- compare agile and more traditional software development, relate lean and agile development, and contrast different agile methodologies for the development of software systems
- use the agile manifest and its accompanying principles
- discuss what is different when leading an agile compared to traditional teams
- explain how software development can be seen as primarily people- and communication-centric
- explain how to scale agile principles and development methods for large organizations
- explain major roles in the agile development process and discuss various levels of authority of team leadership
- describe the SAFe framework
- describe an agile transformation in traditional software companies

Competence and skills

- organize an agile team to maximise its productivity and facilitate communication between people
- collaborate and apply agile way or working in small software development teams
- interact and show progress continuosly with a customer or user
- develop programs using small and frequent iterations in a way that facilitates building in quality from the start
- use test-driven development and test automation, refactor a program and a design, and conduct incremental planning using user stories
- identify and eliminate waste (i.e. unnecessary activities) in the software development process

- design measurement system to follow continuous improvement of teams' performance

Judgement and approach

- explain how software development can be seen as primarily people- and communication-centric
- describe why no single methodology can fit all projects or contexts
- critically assess agile development methods given a certain context and identify common misconceptions
- identify major dysfunctions of the teams and phases they go through during agile transformation- recognize what motivates people for their best performance

Course content

Agile software development aims at setting up an environment to develop software based on the following principles from the agile manifesto:

- Individuals and interactions is valued more than processes and tools
- Working software is valued more than comprehensive documentation
- Customer collaboration is valued more than contract negotiation
- Responding to change is valued more than following a plan

At the core of these principles is the realization that changes are inevitable and the conclusion that change management needs to be integrated into the development process. Agile approaches promotes iterative and incremental development by using a very tight design-code-test cycle.

In this interactive course we will explore together how to apply these agile principles to develop software and manage projects.

The course covers:

- Principles and practices of agile methods such as XP, Scrum and Lean
- Testing and test automation on both unit and system levels
- Communication- and people-centric software development
- Agile methods in relation to more traditional, plan-based methods
- Criticism to agile development methods
- Leading agile development projects and identifying the most important roles
- Scaling agile principles and practices to large organizations
- Measuring key performance indicators (KPIs) of agile teams

Sub-courses

1. **Written hall examination** *(Skriftlig salstentamen),* 3 credits
   Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)
2. **Project** *(Projekt),* 4.5 credits
   Grading scale: Pass (G) and Fail (U)

Form of teaching

The course consists of lectures, and a project part where software is developed using agile approaches.

Language of instruction: English

Assessment

The course is examined by a project. The project is carried out in groups of normally 4-6 students. The project grades are individual based on involving the following:

- The student's active role in the team work.
- An oral presentation of the project status and/or results.
- An individual written report.
- Participation in the project acceptance tests.

The course is also examined by a individual written exam carried out 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).
The grading scale comprises: Pass with Distinction (VG), Pass (G) and Fail (U).

A Pass grade (G) for the entire course requires at least a Pass grade for all sub-courses. A Pass with Distinction grade (VG) for the entire course requires, in addition to at least a Pass grade for all sub-courses, a Pass with Distinction (VG) on the sub-course Written exam.

Course evaluation

The course is evaluated through meeting 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 course DIT191 Agile Development Processes, 7.5 credits. The course cannot be included in a degree which contains DIT191. Neither can the course be included in a degree which is based on another degree in which the course DIT191 is included.