COURSE SYLLABUS

# Product Development in Cross-discipline Teams - 2, 9 credits

*Produktutveckling i interdisciplinära team - Del 2, 9 högskolepoäng*

| | | | |
|---|---|---|---|
| **Course Code:** | TP2S26 | **Education Cycle:** | Advanced level |
| **Confirmed by:** | Dean Mar 1, 2016 | **Disciplinary domain:** | Technology (95%) and social sciences (5%) |
| **Valid From:** | Aug 1, 2016 | **Subject group:** | DT1 |
| **Version:** | 1 | **Specialised in:** | A1F |
| **Reg number:** | JTH 2016/1194-313 | | |

## Intended Learning Outcomes (ILO)

On completion of this course, the student should:

Knowledge and understanding
- show familiarity with development of products that may include software, electronic and mechanical components,
- demonstrate comprehension of practical aspects of Agile product development, including project planning and different activities,
- display knowledge of tools and methods for different activities of project work acquired in the first year courses relevant for the respective area

Skills and abilities
- demonstrate the ability to sustain cross-discipline teams for product development
- demonstrate the ability to deliver intermediate results in an iterative and incremental fashion
- demonstrate the ability to draw a project to a satisfactory conclusion

Judgement and approach
- demonstrate an understanding of different roles in cross-discipline teams during product development
- demonstrate the ability to identify, plan and implement enhancements to an existing product

## Contents

The course supports student teams in the specification and delivery of a software product. The requirement for the product may originate from an external company or organisation, from a need internal to the University, or from an original idea from the students. The software product will be developed through an Agile lifecycle, with clearly defined intermediate deliverable points leading to project closure ready for hand-off to future development or maintenance.
The course covers the following aspects of software delivery:
- Requirements, design and test specification
- Prototype development and unit testing
- Test-driven development (TDD) and paired programming

- Simplest design, continuous redesign and refactoring
- Integration and system testing
- Behaviour-driven development and acceptance testing
- Release planning

## Type of instruction

The course will consist primarily of practical work, supported by supervision meetings and review seminars. Students will work in teams..

The teaching is conducted in English.

## Prerequisites

Passed courses at least 90 credits within the major subject in Computer Engineering, Electrical Engineering (with relevant courses in Computer Engineering), Informatics, Computer Science, Interaction Design (with relevant courses in web programming) or equivalent, and completed courses Product Development in Cross-discipline Teams 1,6 credits and either Software Product Architectures - From Chip to Enterprise, 7,5 credits and Software Product Quality Assurance, 6 credits or User Experience Design, 6 hp and Enterprise Architecture and IT Architecture, 7,5 credits. Proof of English proficiency is required (or the equivalent).

## Examination and grades

The course is graded 5,4,3 or Fail .

The final grade for the course is based upon a balanced set of assessments.
The final grade will only be issued after satisfactory completion of all assesments.

Registration of examination:

| Name of the Test | Value | Grading |
|---|---|---|
| Project work | 8 credits | 5/4/3/U |
| Report and presentation | 1 credit | 5/4/3/U |

## Other information

The product development project will continue work on the code base delivered at the end of Part 1. For situations where this is not possible (for example, exchange students or where teams have had to be reformed), then alternative arrangements will be made.

## Course literature

The literature list for the course will be provided one month before the course starts
Suryanarayana, G., Samarthyam, G., & Sharma, T., Refactoring for software design smells: Managing technical debt., Morgan Kaufmann (2014).
Humble, J., & Farley, D., Continuous delivery: reliable software releases through build, test, and deployment automation., Pearson Education (2010).
Hunt, A., & Thomas, D., The pragmatic programmer: from journeyman to master., Addison-Wesley Professional (2000).