Blekinge Institute of Technology
Department of Software Engineering

Revision: 4
Reg.no: BTH-4.1.1-0303-2018

COURSE SYLLABUS

Evolution och Underhåll av Mjukvara - Projekt

Software Evolution and Maintenance Project

7.5 credits (7,5 högskolepoäng)

Course code: PA2558
Main field of study: Software Engineering
Disciplinary domain: Technology
Education level: Second cycle
Specialization: A1N - Second cycle, has only first cycle course/s as entry requirements

Subject area: Computer Technology
Language of instruction: English
Applies from: 2018-08-01
Approved: 2018-03-01

1. Decision

This course is established by Dean 2016-09-01. The course syllabus is approved by Head of Department of Software Engineering 2018-03-01 and applies from 2018-08-01.

2. Entry requirements

Completed courses of at least 120 ECTS credits of which 90 credits must be in the following areas: Software Engineering, Computer Science. At least 30 credits must be in one or more of the following areas: Programming, Object-oriented Systems, Software Design, Data Structures and Algorithms, Database Technology, Data Communications, Real Time Systems, Operating Systems. In addition, a completed course of at least 7.5 credits in Software Engineering or a Team Software Engineering Project is required.

3. Objective and content

3.1 Objective

Evolution and Maintenance of software is concerned with continuously correcting, adapting, and perfecting the software. This work introduces special challenges such as the need to understand the existing codebase without having access to the original developers, in order to make changes you need to understand the consequences of thee changes, you need to modernise the use of e.g. unit testing, and you need to take time to improve and refactor the codebase according to sound programming principles. It is, however, not enough to understand the source code in itself. You also need to understand how the source code is stored in a configuration management system, and how to suggest, inspect, approve, introduce, and test changes in the software and in the configuration management system. The intention of this course is to, through a practical case, create a deer insight in to challenges such as the above.

3.2 Content

- Understanding of an existing software system
- Testing of software
- Debugging of Software
- Time- and Size Estimates of Software Maintenance
- Evolution
- Software Maintenance
- Configuration Management
- Working with an Issue Tracker
- Understanding of the importance of clean code.

4. Learning outcomes

The following learning outcomes are examined in the course:

4.1 Knowledge and understanding

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

- be able to present an overview of the challenges with software evolution
- in detail be able to present the challenges involved in software maintenance.

4.2 Competence and skills

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

• be able to work with a modern development environment, with an IDE, testing tools, and configuration management tools.
• be able to understand and debug legacy software.
• be able to estimate size and time to introduce and test changes into legacy software.
• be able to introduce controlled changes in legacy software to address concrete bug reports.
• be able to test the introduced changes in a controlled way.

4.3 Judgement and approach

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

• be able to analyse source code for improvements on code, design, and architecture.
• be able to document and reflect on performed work.

5. Learning activities

The course is run as a development project, mixed with lectures and seminars.

6. Assessment and grading

Modes of examinations of the course

Code	Module	Credits	Grade
1810	Project assignment	6 credits	GU
1820	Written report	1.5 credits	AF

The course will be graded A Excellent, B Very good, C Good, D Satisfactory, E Sufficient, FX Fail, supplementation required, F Fail.

The course information for each course revision should include the assessment criteria and make explicit in which modes of examination that the learning outcomes are assessed.

7. Course evaluation

The course evaluation should be carried out in line with BTH:s course evaluation template and process.

8. Restrictions regarding degree

The course can form part of a degree but not together with another course the content of which completely or partly corresponds with the contents of this course.

9. Course literature and other materials of instruction

Priyadarshi Tripathy, Kshirasagar Naik, "Software Evolution and Maintenance: A Practitioner's Approach ", Wiley, 2015. ISBN: 978-0-470-60341-3

R. Martin "Clean Code", Prentice Hall; 1 edition, 2008. ISBN-13: 978-0132350884, ISBN-10: 0132350882

Reference Literature:
S. McConnel "Code Complete", Microsoft Press; 2nd edition, 2004. ISBN-13: 978-0735619678 , ISBN-10: 0735619670

J. Richardson "Ship it!", the Pragmatic Bookshelf, 2005.