Blekinge Institute of Technology
Department of Software Engineering

COURSE SYLLABUS

Programvaruarkitektur och kvalitet

Software Architectures and Quality

7,5 ECTS credit points (7,5 högskolepoäng)

Course code: PA1453
Educational level: First cycle
Course level: G1F
Field of education: Technology
Subject group: Computer Technology

Subject area: Software Engineering
Version: 5
Applies from: 2017-08-28
Approved: 2017-02-20

1 Course title and credit points
The course is titled Software Architectures and Quality/Programvaruarkitektur och kvalitet and awards 7,5 ECTS credits. One credit point (högskolepoäng) corresponds to one credit point in the European Credit Transfer System (ECTS).

2 Decision and approval
This course is established by Dean 2017-02-02. The course syllabus was revised by Head of Department of Software Engineering and applies from 2017-08-28.
Reg.no: BTH-4.1.1–0022-2017.
Replaces: PA1410.

3 Objectives
Software Architecture is an important technical concept in modern large-scale software development that serves several purposes: planning development resources; analyzing the problem domain from multiple perspectives using the architectural views; and abstracting big amounts of information to obtain a useful overview of the system.
In addition, the architecture and the decisions on how to construct architecture are a key component to plan and achieve a certain level of quality in a system, which thereby determines how successful the system will be.
In the course the students are expected to obtain detailed knowledge about software architectures and software quality and the effects software architectures and architectural decisions have on the qualities of the software product under development.
The students will acquire understanding about how to: i) document software architectures of existing systems; ii) construct software architectures based on modern methods and ideas such as architecture styles and patterns and tactics, by considering the intended product, the surrounding technology and

the developing organization; iii) discuss the suitability of a software architecture for a specific software product in a specific context using facts as basis.

4 Content
The course comprises the following topics:
•Quality aspects in software and software architectures.
•Architectural, views, languages styles, patterns and tactics.
•Recovery and description of software architecture from source code.
•Methods for architectural design, evaluation and transformation.
•Literature review of relevant research articles about software architecture.

5 Aims and learning outcomes
Knowledge and understanding
On completion of the course, the student should be able to:
•Read, interpret, produce and reason about software architecture descriptions, relating to different architectural concerns, with some of the most common architectural views.
•Recognize and reflect about the connections between software architecture, product qualities, the development process, the resource needs, and the development organization.
Competence and skills
On completion of the course, the student should be able to:
•Produce, document and communicate a simple system software architecture, so that this documentation can be used as guidance to develop the system and to achieve the desired qualities/characteristics.
•Perform trade-offs among quality requirements and design decisions, apply the suitable solutions to achieve the quality requirements.

•Identify and document the implementation rules and guidelines based on a software architecture.
Judgement and approach
On completion of the course, the student should be able to:
•Evaluate a software architecture, and with facts an explicit assumption as the basis, assess how well the development work needs are met, and the quality characteristics can be achieved.
•Assess and criticize, in architectural terms, the architectural decisions in software architectures.
•Analyze, recommend and apply architectural modifications and improvements to be able to meet certain quality levels.

6 Learning and teaching
The course builds upon individual readings, lectures, individual research work, as well as team work in small projects (assignments).
The course contents are organized around a number of lectures where the students are expected to actively participate by discussing, questioning and reflecting about their own experiences, and their understanding of the conflicting concepts they have read in the course literature and in research papers in the area. These lectures together with their readings constitute the pillars over which the group assignments build upon.
The two assignments consist on real-world but scale-controlled exercises in which the students are intended to apply the concepts, methods and techniques presented in class to document, evaluate and eventually transform software architectures. Interleaved in between the lectures and assignments, there are a number of practical sessions (seminars) in which the students, with the assistance of the instructor, cooperate to solve some of the problems related to their own assignments.
A written exam completes the course, comprising both multiple-choice and / or essay, open questions. This gives a mixture of state-of-the-art as described in research articles, and state-of-practice as evidenced by personal experiences, and industry studies.
English

7 Assessment and grading
Examination of the course

```
-------------------------------------------------
Code Module                 Credit      Grade
-------------------------------------------------
1710 Assignment 1           1.5 ECTS    A-F
1720 Assignment 2           3 ECTS      A-F
1730 Presentation and opposition 1 ECTS      G-U
1740 Exam                   2 ECTS      A-F
-------------------------------------------------
```

The course will be graded A Excellent, B Very good, C Good, D Satisfactory, E Sufficient, FX Fail, supplementation required, F Fail.The final course grade is the aggregate of these four examination parts. The course grade is set when all the examination parts are approved.

8 Course evaluation
The course coordinator is responsible for systematically gathering feedback from the students in course evaluations and making sure that the results of these feed back into the development of the course.

9 Prerequisites
The student must have successfully completed a total of 22,5 ECTS credits within the following areas: Software Design, Data Structures and Algorithms, Database Systems, Computer Networks, Real-Time Systems, Operating Systems and successfully completed a course in Programming of at least 6 credits.

10 Field of education and subject area
The course is part of the field of education and is included in the subject area Software Engineering.

11 Restrictions regarding degree
The course cannot form part of a degree with another course, the content of which completely or partly corresponds with the contents of this course.

12 Course literature and other teaching material
Main Books:
1. Software Architecture in Practice, 3rd Edition
Authors: L. Bass, P. Clements, and R. Kazman
Addison-Wesley, 2012.
ISBN10: 0321815734 ISBN13: 97803218157362.
2. Applied Software Architecture
Authors: C. Hofmeister, R. Nord, and D. Soni, Addison-Wesley Professional, 2000.
ISBN 10: 0321643346 ISBN 13: 9780321643346.
Reference Materials:
3. Documenting Software Architectures: Views and Beyond, 2nd Edition
Authors P. C. Clements et al.,
Pearson Education, 2010.
ISBN 10: 0321552687 ISBN 13: 978-0321552686.
∎