

# **Course Program**

## **ETS170 Requirements Engineering**

**<http://cs.lth.se/ets170>**

**Björn Regnell, Elizabeth Bjarnason**

**Study period: 2016-HT2, Revision date: October 11, 2016**

The objective of the course is to give basic and advanced knowledge and skills within requirements engineering for large-scale development of systems completely or partly based on software. The course gives both theoretical knowledge and practical skills in methods and techniques for requirements engineering. The course gives training in scientific paper reading.

## **1 Learning Objectives**

### **1.1 Knowledge and understanding**

For a passing grade the student must

1. be able to define basic concepts and principles within requirements engineering
2. give an account of several different types of requirements
3. be able to describe and value several different methods and techniques for requirements engineering
4. be able to describe and relate different sub-processes within requirements engineering
5. be able to describe the relation between the requirements engineering process and other processes in the product lifecycle
6. be able to describe the relation between requirements engineering and market-driven product management
7. be able to discuss some scientific results within requirements engineering research

### **1.2 Skills and abilities**

For a passing grade the student must

1. be able to choose suitable requirements techniques for a given context
2. be able to apply several different techniques for requirements elicitation
3. be able to apply several different techniques for requirements specification
4. be able to apply several different techniques for requirements validation
5. be able to apply several different techniques for requirements prioritisation

### **1.3 Judgement and approach**

For a passing grade the student must

1. be able to consciously select a process depending on the nature of the requirements
2. show a systematic and long-term approach to processes
3. be able to consciously see the problem in the relation between the quality of requirements and the quality of the resulting implementation
4. be able to adequately involve users in the requirements engineering process
5. be able to consciously see the problem in the relation between requirements engineering and economical aspects of product development

## **2 Contents**

The course includes theory and practice regarding the following topics:

1. Requirements on different abstraction levels and in different contexts
2. Sub-processes of requirements engineering and their relation
3. Specification of data requirements, e.g. using virtual windows and data models
4. Specification of functional requirements, e.g. using textual feature requirements and task descriptions
5. Specification of different types of non-functional requirements, e.g. usability, performance, reliability
6. Different techniques for requirements elicitation
7. Different techniques for requirements validation
8. Different techniques for requirements prioritization
9. Market-driven requirements engineering and product management

### 3 Course elements

- L: Lectures** The lectures provide an overview of the literature. They do not cover every detail, but give a high-level structure of the subject and thereby aid self-studies of the literature. Discussions are promoted.
- E: Exercises** The main objective of the exercises is to support the project and prepare for the written exam through prototypical problems, by connecting theory to practice and to give opportunity to discuss details of RE techniques.
- LAB: Computer lab sessions** The lab sessions illustrate computer supported prioritization and release planning, and demonstrates the complexity of requirements selection and scheduling. Preparations are mandatory.
- P: Project** The project is carried out in groups of 6-8 students. The project involves a number of deliverables and a final project conference where the learning outcome of each project is presented. Project groups are established during the first course week. Project assignments are established during the second course week.

### 4 Assessment

- The project is graded fail/3/4/5 based on project deliverables.
- Approved lab session preparations and assignments are required for passing.
- The written exam comprises 100 points. The pass limit is 50 points.
- It is optional to hand in two sets of candidate exam problems. Good candidate exam problems can give a maximum of 10 bonus points added to the result of the written exam if they are handed in before stipulated deadlines. The bonus can be used at written exams within 10 months after the course ending.
- The final course grade on the scale fail/3/4/5 is based on the written exam points including bonus and the project grade using the following mapping:

	Project: 3	Project: 4	Project: 5
	Exam points + bonus		
Final: 3	≥ 50	≥ 50	≥ 50
Final: 4	≥ 75	≥ 67	≥ 60
Final: 5	≥ 90	≥ 83	≥ 75

### 5 Literature

The course elements and the written exam will cover the following literature:

- Lau Soren Lauesen, Software Requirements - Styles and Techniques, Addison-Wesley, ISBN 0-201-74570-4, 2002.
- LAB1&2 Preparations and instructions for Lab 1: "Requirements Modeling" and Lab 2: "Requirements Prioritization and Release Planning"
- MDRE "Market-Driven Requirements Engineering for Software Products", Björn Regnell and Sjaak Brinkkemper, Engineering and Managing Software Requirements, Eds. A. Aurum and C. Wohlin, Springer, ISBN 3-540-25043-3, 2005
- PRIO "Requirements Prioritization", Patrik Berander and Anneliese Andrews, Engineering and Managing Software Requirements, Eds. A. Aurum and C. Wohlin, Springer, ISBN 3-540-25043-3, 2005
- OSSRE "Understanding Requirements for Open Source Software", Walt Scacchi, Design Requirements Engineering: A Ten-Year Perspective, Springer, pp. 467-494, 2009
- RP "The Art and Science of Software Release Planning", Günther Ruhe and Moshood Omolade Saliu, IEEE Software, November/December, pp. 47-53, 2005
- QUPER Supporting Roadmapping of Quality Requirements - Björn Regnell, Richard Berntsson Svensson, Thomas Olsson, IEEE Software 25(2) pp 42-47 March-April 2008
- INSP "Att inspektera krav". Sid 67-76, Framgångsrik kravhantering, andra utgåvan, Teknikföretagen, Joachim Karlsson, V040072, ISSN 1103-7008, 1998
- INTDEP "An industrial survey of requirements interdependencies in software product release planning", Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B., Natt och Dag, J.: Int. Conf. on Requirements Engineering (RE01), Toronto, Canada, pp. 84-91, 2001
- AGRE "Agile Requirements Engineering Practices: An Empirical Study", Lan Cao, Balasubramaniam Ramesh, IEEE Software, January/February 2008, pp.60-67, 2008

## 6 Overview

	<i>Topic</i>	<i>Literature</i>	<i>When</i>
<b>Week 1</b>	L1 Introduction BR	Lau:1	Mon 10-12
	E1 Requirements types	Lau:1	Mon 15-17 or Wed 8-10
	L2 Elicitation, Specification 1 BR	Lau:8,2-3.5	Th 10-12
	P Project Mission v1 -> BR EB AO		Deadline Fri 09:00
<b>W2</b>	L3 Spec 2, Mission lottery AO	Lau:3.6-4	Mon 10-12
	E2 Elicitation	Lau: 8	Mon 15-17 or Wed 8-10
	P Meeting with supervisor		
	reqT reqT & Scala tutorial - BR	reqT.org	Thu 10-12
<b>W3</b>	P Project Mission v2 -> supervisor		Mon 09:00
	L4 Market-driven + Open Source RE, Prio, Rel Plan JL	MDRE, PRIO, RP, OSSRE	Mon 10-12
	E3 Functional requirements	Lau:2-4	Mon 15-17 or Wed 8-10
	Lab1 Requirements Modelling	LAB1	Wed 13-15 or 15-17 or Thu 8-10 or 10-12
<b>W4</b>	P Release R1 -> supervisor		Deadline Mon 09:00
	P Meeting with supervisor		
	L5 Specification 3, Quality, Lifecycle	Lau:5-7, QUPER	Mon 10-12
	E4 Quality requirements	Lau:6, QUPER	Mon 15-17 or Wed 8-10
	GL Guest: Hampus Jacobsson		Th 10-12
<b>W5</b>	L6 Validation, Inspections, Interdependencies, Agile RE	Lau:9, INSP, INTDEP, AGRE	Mon 10-12
	E5 Validation	Lau:9, INSP	Mon 15-17 or Wed 8-10
	Lab2 Prioritization, Release Planning	LAB2	Wed 13-15 or 15-17 or Thu 8-10 or 10-12
<b>W6</b>	P Release R2 -> customer, supervisor		Deadline Mon 09:00
	P Meeting with supervisor		
	P Validation Checklist-> customer & both teams' supervisors		Deadline Mon 09:00
	P Validation Report-> developer & both teams' supervisors		Deadline Fri 09:00
	P Conference presentation -> Johan, Alma, Eliza		Deadline Sun 15:00
<b>W7</b>	L7 Project conference		Mon 10-12
	P Release R3 -> supervisor		Deadline Sun 23:59
<b>Jan</b>	Exam	All literature	January 12, Thu 14-19
<b>Feb</b>	P Course Evaluation -> Björn, Eliza, 1 per project		Jan 30, Mon 09:00

## 7 Personnel

Bjorn.Regnell BR  
 Elizabeth.Bjarnason EB  
 Alma.Orucevic-Alagic AO  
 Johan.Linaker JL  
 Daniel.Helgesson DH  
 Lena.Ohlsson LO  
 ...all e-mails @cs.lth.se

Lectures  
 Coordinator, Exercises, Exam  
 Lectures, Exam, Project and Lab supervision  
 Lectures, Project supervision, Lab coordinator  
 Project supervision, Lab supervision  
 Course Secretary

## 8 Rooms

<i>What</i>	<i>Where</i>
Lectures	Genetikhusets aula (W1 Mo), MA:MA03 (W2- Mo), E:C3 (W1, W2 Th)
Exercises	E:3336 / E:1147
reqT	MA:MA05 (Thu, W2)
LAB	E:Varg (Wed, Thu, W3, W5)
Exam	MA:MA010-H- I

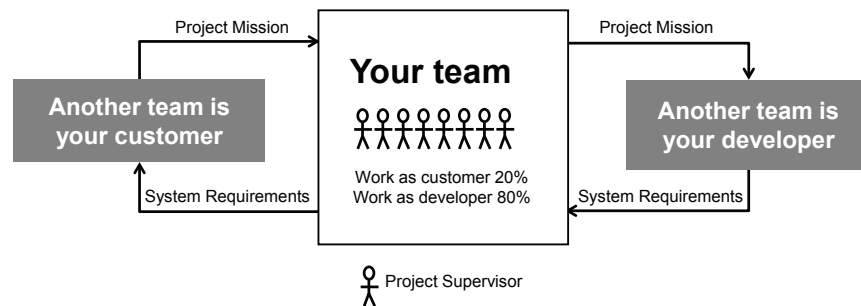
## 9 Project

### 9.1 Objectives

The main goals of the project from a course perspective are to:

1. connect theory to practice,
2. give a concrete experience of practical requirements engineering,
3. promote student motivation through real stakeholders, and
4. provide a group-learning setting that is focused on realistic problems.

### 9.2 Context and roles



Each project team has two different assignments:

**Customer work** (Kundarbete) To invent Project Mission v1 and act as customer providing domain expertise and feedback to another team that acts as developer.

**Development work** (Utvecklingsarbete) To develop a system model including requirements of different types at appropriate abstraction levels based on a given Project Mission from another team acting as your customer.

The project team consists of 6-8 members and these managers should be appointed among team members:

**P3RM** Project, Process, Prioritization, and Release Manager (1 person)

**SCCVM** Stakeholder, Customer Communication and Validation Manager (1 person)

**TDEVM** Tools, Documents, Experiences and Version Manager (1-2 persons)

**EPM** Elicitation and Prototyping Manager (1-2 persons)

**QRM** Quality Requirements Manager (1 person)

**DRM** Data Requirements Manager (1 person)

The manager roles imply management, planning and coordination responsibilities, but managers should not do all the work: *all members should contribute in all parts!*

### 9.3 General project rules

1. The project comprises 80 hours per person.
2. Approximately 80% of each team member's total effort should be devoted to development work.
3. Approximately 20% of each team member's total effort should be devoted to customer work.
4. The total effort should be evenly distributed among participants.
5. In weeks W2, W4, and W6 a meeting should be scheduled with the project supervisor, where the project team reports on status, challenges and plans.

### 9.4 Project deliverables

Phase	Deliverables	Deadline
Definition	Project Mission v1	Week 1: Friday 09:00
Planning	Project Mission v2	Week 3: Monday 09:00
Iteration 1	Release R1	Week 4: Monday 09:00
Iteration 2	Release R2	Week 6: Monday 09:00
Iteration 3	Validation Checklist	Week 6: Monday 09:00
	Validation Report	Week 6: Friday 09:00
	Conference Presentation	Week 6: Sunday 15:00
	Release R3	Week 7: Sunday 23:59
	Course Evaluation	Jan 30, Monday 09:00

All deliverables should have a title, version number, team id (capital letter), system name and names of the project members.

#### 9.4.1 Project Mission v1

Acting as customer your team should prepare an initial version of a Project Mission for a development team. The Project Mission defines the system for which the employed project will elicit, prioritize, specify and validate requirements.

1. The Project Mission should fit on **one A4 page**, be in .pdf format for easy printing and web publishing.
2. The Project Mission should include a descriptive project name as well as the names of its authors.
3. You should fulfill the following criteria regarding the system you propose:
  - (a) You have a deep understanding of the application domain
  - (b) You have a genuine interest in the system
  - (c) You are able to assess the value of detailed requirements

4. In the Project Mission you should state which one of the following customer roles that you would like to play:

**Key Customer** Your team acts as one of many potential customer on an assumed market for the envisioned product. The development team owns the product and decides on priorities and product content, while your team gives input and provides feedback.

**Product Owner** You as a customer team is planning to sell the product yourself in an envisioned future. The development team is subcontracted to do requirements engineering only. Your team decides on priorities and scope.

### 9.4.2 Project Mission v2

Acting as developer your team should prepare a second version of the Project Mission where the scope of the project is further defined in dialog with your customer and supervisor.

1. The Project Mission v2 is recommended to include the following information:
  - (a) Table of contents
  - (b) Background and other information from Project Mission v1
  - (c) Main goals and system context
  - (d) Participants and potential stakeholders
  - (e) Description of planned activities and deliverables with deadlines
  - (f) Diagram showing the planned time per week and participant
  - (g) Responsibilities of project members
2. With the above content it is useful if following questions can be answered:
  - (a) What is the project about?
  - (b) Who is participating in the project as members and as input providers?
  - (c) What should be done in the project?
  - (d) When should the results be delivered?
  - (e) Who is responsible for what?
  - (f) When shall who work with what?
  - (g) What will the customer team spend their time on?

### 9.4.3 Deliverables

1. You should work iteratively and divide your work into 3 main iterations, each ending with a release with all your accumulated work products. (You may have more sub-iterations with additional team-internal releases.)
2. The releases are denoted R1, R2 and R3.
3. For each release, the quality of your deliverables should represent a noticeable improvement.
4. Each release should be divided into two explicit parts: **System Requirements** and **Project Experiences**, each with its own **table of contents**.
5. There should be an **overview description** of each release to make navigation and assessment easy, e.g. in a file called `index.html` or `README.txt`.
6. A release Rn of team X should be delivered in *one single, self-contained zip-file* named `X-Rn.zip` including *all* deliverables. If the file is too big to email then provide a http link to a downloadable zip-file.
7. Each deliverable may link to further resources such as html pages, pdf documents, screen images, text files, executables, etc., all contained in the delivered zip file. No external links outside the zip is allowed.
8. The last release R3 should include final versions of: System Requirements, Project Experiences, Validation Report & Checklist (final versions by R2 also copied into R3), and Conference Presentation. Course Evaluation is delivered post course.

**System Requirements** including the following information:

- (a) Different types of system requirements (e.g. data, function, quality) at different levels (e.g. goal, domain, product, design).
- (b) Several different specification techniques (e.g. context diagrams, features, virtual windows, task descriptions).
- (c) Each requirement should have a unique identity (name or number).
- (d) A subset of the requirements should be prioritized.
- (e) A subset of the requirements should be implemented as mock-up designs (e.g. screens and prototypes, analog drawings, clickable presentations, executable gui mockups).
- (f) A subset of the requirements should be release planned. The release planning information should define which requirements that are implemented by the development team as mock-up designs in release R3, and which requirements are selected to be fully implemented in the imagined releases R4 and R5.

**Project Experiences** including the following information:

- (a) a description of your requirements engineering work, including experiences and reflections in relation to learning objectives.
- (b) Description of the chosen methods/techniques for elicitation, specifi-

cation, validation, and prioritization.

- (c) Motivation for *why* you chose the used methods/techniques.
- (d) Reflection on the usage of these methods/techniques in terms of what was successful and what was challenging. Example questions for reflection: What have you learned in relation to the learning objectives in this course program? What would you have done differently if you would do this project again as a "real" project, based on what you know now? What have you learned in relation to the learning objectives?
- (e) A personal statement by each team member that briefly explains each individual's contributions to the project results.
- (f) The Project Experiences should *not* include course evaluation issues, but focus on your own work and learning outcome.

**Validation Report** Acting as customer, you should validate release R2 from the other development team and hand in your validation report together with your team's R2. Your team should produce relevant and useful issues for improvement. Each issue should be ranked for criticality.

**Validation Checklist** Acting as developer you should provide your customer with a requirements validation checklist tailored to the context.

**Conference Presentation** Prepare and rehearse a short presentation.

- (a) The total presentation time and further guidelines are given during the course.
- (b) Spend approx. 10% of the presentation time on the project's mission.
- (c) Spend approx. 45% of the time on project results and techniques used.
- (d) Spend approx. 45% of the time on experiences and learning outcome.
- (e) Slides should be in { .ppt|.pptx|.pdf }.

**Course Evaluation** (Not part of assessment.) A separate, free-form Course Evaluation document should be handed in by the team. If team members have different views, it is valuable if these differences are reflected. For each relevant course element (L, E, LAB, P, etc.) answer questions such as: What worked well? If something needs improvement, *why* and *how* would you like it to be changed?

## 9.5 Project assessment

1. The deliverables Project Mission and Conference Presentation is pass/fail only.
2. The project grade of fail/3/4/5 is based on Release R3 and your Validation Report & Checklist according to the criteria in the table on the next page:

Assessment area	Requirements for project grade 3 Demonstrate <b>acceptable</b> ability to ...	Also required for project grade 4 Demonstrate <b>advanced</b> ability to ...	Also required for project grade 5 Demonstrate <b>excellent</b> ability to ...
<b>Specification</b>	<b>3A)</b> apply more than one suitable specification technique (e.g. task descriptions and screen prototypes), and more than two types of requirement (e.g. data, function, quality), and more than three abstraction levels (e.g. goal, domain, product, design). <b>3B)</b> define a system's boundaries and its interaction with external entities. <b>3C)</b> reflect on specification experiences and reason about choices of specification methods in relation to different contexts.	<b>4A)</b> combine different degrees of completeness and different levels of abstraction. <b>4B)</b> use at least four different specification techniques adequately tailored to the context. <b>4C)</b> provide explicit requirements rationale that reduce risks of misinterpretation. <b>4D)</b> use hierarchies and requirements relations to manage evolving requirements structures.	<b>5A)</b> combine specification techniques in an explicitly motivated trade off between qualities and costs, where a high degree of specification completeness is achieved for a carefully selected subset of requirements. <b>5B)</b> provide motivated estimations of target quality levels using well-defined scales.
<b>Elicitation</b>	<b>3D)</b> apply more than one elicitation technique in a relevant way. <b>3E)</b> reflect on elicitation experiences.	<b>4E)</b> reason about the need for further elicitation in relation to specification quality.	<b>5C)</b> go beyond initial stakeholders and given frames, while challenging the domain boundaries and eliciting creative ideas and deep domain knowledge in real-world contexts.
<b>Validation</b>	<b>3F)</b> to assess the quality of requirements and find relevant problems of several different types. <b>3G)</b> apply more than one validation technique. <b>3H)</b> reflect on validation experiences.	<b>4F)</b> to find, prioritize and discuss requirements quality problems of different types, while reaching beyond form issues. <b>4G)</b> adapt the validation to the context and provide rationale for the chosen validation techniques.	<b>5D)</b> reason about the relation between requirements quality problems and risks, both from a customer and developer viewpoint. <b>5E)</b> utilize links among different types of specifications in validation efforts to find and address potentially harmful inconsistencies.
<b>Prioritization</b>	<b>3I)</b> use more than one prioritization technique in a relevant way. <b>3J)</b> reflect on prioritization experiences.	<b>4H)</b> create a release plan for a subset of prioritized features, while taking into account precedence constraints.	<b>5F)</b> combine priorities from several stakeholders and use priorities and scheduling constraints to iteratively create a relevant release plan. <b>5G)</b> use prioritization to focus improvements of specification quality and elicitation efforts for a well motivated subset of requirements.