Project Description TFRG55 Requirements Engineering for Digital Systems http://cs.lth.se/krav

Björn Regnell, Elizabeth Bjarnason, Johan Linåker

Study period: 2022-VT1, Revision date: January 5, 2022

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.

2 General project rules

1. The project comprises 80 hours per person.
2. In weeks W2, W4, and W6 a group supervision meeting is held, where you report on status, challenges and plans, and discuss these with the supervisor and with the other students.

3 Project deliverables

Phase	*Deliverables*	*Deadline*
Planning	Project Mission v2	Week 2: Tuesday 23:59
Iteration 1	Release R1	Week 3: Sunday 23:59
Iteration 2	Release R2	Week 5: Sunday 23:59
	Validation Checklist	Week 5: Sunday 23:59
	Validation Report	Week 6: Thursday 23:59
Iteration 3	Release R3	Week 7: Sunday 23:59
	Course Evaluation	March 31st, Thursday 23:59

All deliverables should have a title, version number, system name and name of submitter.

3.1 Project Mission v2

You should prepare a second version of the Project Mission where your initial understanding of the scope of the project is described.

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, including a context diagram
   (d) Participants and potential stakeholders
   (e) Description of planned activities and deliverables with deadlines
2. With the above content it is useful if the following questions can be answered:
   (a) What is the project about?
   (b) What should be done in the project?
   (c) When should the results be delivered?

3.2 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 internal releases.)
2. The releases (delivered for the course) 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.
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 are allowed.
8. The last release R3 should include final versions of: System Requirements, Project Experiences, Validation Report & Checklist (final version by R2 and included in R3). Course Evaluation is delivered post course.

System Requirements includes the following:

(a) Different types of system requirements (e.g. data, function, quality) at different levels (e.g. goal, domain, product, design).
(b) Several specification techniques (e.g. context diagrams, features, virtual windows, task descriptions).
(c) Each requirement should have a unique identity (name or number).
(d) Design-level requirements are to be specified for the most prioritized requirements. This sub-set of requirements shall be implemented as mock-up designs in the final course delivery (R3) using, e.g. screens and prototypes, analog drawings, clickable presentations, executable GUI mockups.

Project Experiences include the following:

(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, specification, 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 based on what you know now? What have you learned in relation to the learning objectives?
(e) The Project Experiences should *not* include course evaluation issues, but focus on your own work and learning outcome.

Validation Report To gain experience and input to your own project, you will validate another project release R2 and hand in your validation report together with your R3. You should produce relevant and useful issues for improvement. Each issue should be ranked for criticality.

Validation Checklist To help another project to validate your release R2, you will provide them with a requirements validation checklist tailored to the context.

Course Evaluation (Not part of the assessment.) A separate, free-form Course Evaluation document should be handed in by the team. For each relevant course element (lectures, exercises, labs, project etc) answer questions such as: What worked well? If something needs improvement, *why* and *how* would you like it to be changed?

4 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 **acceptable** ability to ...	*Also required for project grade* **4** Demonstrate **advanced** ability to ...	*Also required for project grade* **5** Demonstrate **excellent** ability to ...
Specification	**3A)** 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). **3B)** define a system's boundaries and its interaction with external entities. **3C)** reflect on specification experiences and reason about choices of specification methods in relation to different contexts.	**4A)** combine different degrees of completeness and different levels of abstraction. **4B)** use at least four different specification techniques adequately tailored to the context. **4C)** provide explicit requirements rationale that reduce risks of misinterpretation. **4D)** use hierarchies and requirements relations to manage evolving requirements structures.	**5A)** 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. **5B)** provide motivated estimations of target quality levels using well-defined scales.
Elicitation	**3D)** apply more than one elicitation technique in a relevant way. **3E)** reflect on elicitation experiences.	**4E)** reason about the need for further elicitation in relation to specification quality.	**4F)** demonstrate good use of prototyping to elicit realistic user requirements. **5C)** go beyond initial stakeholders and given frames, while challenging the domain boundaries and eliciting creative ideas and deep domain knowledge in real-world contexts.
Validation	**3F)** to assess the quality of requirements and find relevant problems of several different types. **3G)** apply more than one validation technique including prototyping. **3H)** reflect on validation experiences.	**4G)** to find, prioritize and discuss requirements quality problems of different types, while reaching beyond form issues. **4H)** adapt the validation to the context and provide rationale for the chosen validation techniques.	**5D)** reason about the relation between requirements quality problems and risks, both from a product owner and developer viewpoint. **5E)** utilize links among different types of specifications in validation efforts to find and address potentially harmful inconsistencies.
Prioritization	**3I)** use more than one prioritization technique in a relevant way. **3J)** reflect on prioritization experiences.		**5G)** use prioritization to focus improvements of specification quality and elicitation efforts for a well-motivated subset of requirements.