

Context Project Guidelines 2012/13

(for students)

Introduction

The goal of the context project is to perform a software project specified by an external party in a given context. Students will experience theories of software engineering and teamwork by applying them to a concrete case. This includes requirements engineering, architecture and software design as well as the implementation of the software system following an agile software engineering process. The projects will be finished with a presentation and demonstration of developed software systems.

Learning objectives

The context project has the following process, presentation, and product specific learning objectives. After completing this course the student will be able to:

Process

- Work in a team following an iterative and incremental software development process, as determined by the group
- Work together with users and stakeholders from a non-IT context
- Analyze and evaluate a problem in a non-IT context
- Acquire the information and knowledge on the given non-IT context from the literature and experts in order to design and develop the solution
- Evaluate the requirements/user stories of different stakeholders
- Develop the system following an agile software development process (plan each iteration of the project, reflect and evaluate the planning regularly, and adjust the planning to changes in the project and the environment)
- Reflect on his/her contribution to the project and the final product
- Form a vision of the role of IT in the given context and its role for solving the given problem

Product

- Analyze and document the system requirements in a requirements document
- Design and document the architecture for the given problem in a non-IT context
- Specify an iterative and incremental test and implementation plan
- Develop a prototype and/or final product according to the requirements, design, implementation and test plan
- Continuously test and evaluate the solution

Presentation

- Report in oral form to his/her peers, supervisors, users and customers
- Report in written form to his/her peers, supervisors, users and customers

Context projects and teachers for 2012/13

The following context projects are available:

1. Health Informatics / Medical Informatics (Willem-Paul Brinkman)
2. Crisis Management: Improving Resilience to Incident Management, Search & Rescue Missions, and Public Safety (Catholijn Jonker, Koen Hindriks)
3. Mijn Cultureel Erfgoed (Alan Hanjalic)
4. Programming Life: Synthetic Biology (Dick de Ridder, Marcel Reinders)
5. Computer Games (Rafael Bidarra)

List of deliverables

During the project, the teams need to submit a number of deliverables that will be used for the assessment. Basically, for each deliverable the team first submits a draft version to the teaching assistants. They will provide feedback on the draft that then needs to be integrated by the students. The group submits the final version, which will be considered for the grading.

General guidelines for the documents:

For each document, the following requirements apply. Each document should contain:

- Title page (title of the document, name and id of the students, date, version number)
- Short abstract about the contents of the document
- Table of contents
- Document specific content
- References to literature and other sources of information

List of deliverables and schedule:

The list of the deliverables and corresponding deadlines for the submission of the draft and final versions is:

- Product vision and planning (draft: Q3/W4, initial: Q3/W5, final: Q4/W8)
- Emergent architecture design (drafts: Q3/W5, Q4/W2, Q4/W6; final: Q4/W8)
- Context seminar report (draft: Q3/W6, final: Q3/W8)
- Peer reviews (Q3/W8 and Q4/W8)
- Lightweight SCRUM plans (before each iteration)
- Final report (draft: Q4/W7, final: Q4/W8)
- Final product (version for SIG: Q4/W2, final code for SIG: Q4/W8)
- Final presentation (Q4/W9, exact date will be announced)

The exact deadlines will be posted on Blackboard and will be strictly adhered to. Not handing in documents in time may lead to exclusion from the project.

The following provides a guideline on how to structure each deliverable (might be adapted for a particular context). Page numbers refer to "Document specific content", excluding title page, abstract, table of contents, and references.

Product vision (max. 8 pages A4)

1. Introduction
2. Product
 - 2.1. Product vision
 - 2.2. High-level product backlog (set of epics aligned with the vision)
 - 2.3. Roadmap (major release schedule, release goals)
3. Product backlog (first version with estimates and prioritized user stories)

- 3.1. User stories of features
- 3.2. User stories of defects (if applicable)
- 3.3. User stories of technical improvements (if applicable)
- 3.4. User stories of know-how acquisition
- 3.5. Initial release plan (milestones, MRFs per release)
4. Definition of Done (backlog items, sprints, releases)
5. Glossary

For an overview of what the document should contain, consult Kenneth S. Rubin, "[Essential Scrum: A Practical Guide to the Most Popular Agile Process](http://proquest.safaribooksonline.com/)" (login first at <http://proquest.safaribooksonline.com/>), in particular chapters 15 and 17. An example document will be posted on Blackboard.

Emergent architecture design (final version max. 10 pages A4)

Note that this document needs to be updated during the various sprints to present the current state of the design. The initial version can contain sketches/ideas of the architecture design; the document is then continuously extended throughout the sprints. An example document will be posted on Blackboard.

1. Introduction
 - 1.1. Design goals
2. Software architecture views
 - 2.1. Subsystem decomposition (sub-systems and dependencies between them)
 - 2.2. Hardware/software mapping (mapping of sub-systems to processes and computers, communication between computers),
 - 2.3. Persistent data management (file/ database, database design)
 - 2.4. Concurrency (processes, shared resources, communication between processes, deadlocks prevention)
3. Glossary

Peer reviews

In the fourth week of each quarter the students perform an online peer-review of all team members in the project group including themselves. The review contains for each team member an evaluation of:

- The know-how the team member is bringing into the group
- The way of working
- The collaboration with the other team members

The content of the reviews will not be used for grading, but help give the teachers insight into the group development process. Not taking the reviews seriously will however result in a lower grade for the process score.

Lightweight SCRUM Plans (before each iteration)

Iterations in the implementation phase of the system last for two weeks. Before each iteration of the implementation of the system, the project group submits a plan to the teaching assistant. The planning comprises:

- The selection of a set of backlog items (important items first)
- A list of the tasks for each selected item
- The assignment of students to tasks
- An estimation of the effort per task
- The actual effort per task (after the iteration is done)
- Short reflection on the main problems and adjustments of the iteration planning

The planning of the previous iteration is used as input for the planning of the next iteration. At the end of the project the plans are added as an appendix to the final report. The students are free to choose the template for the planning.

Final report

1. Introduction
2. Key problem(s) and solution(s) – highlights
3. Reflection on the teamwork
4. Individual reflection on the project (max. 1 A4 per team member)
5. Lightweight SCRUM Plans

Final product

The source code should be well structured and include unit tests. The quality of the source code (including test code) will be evaluated by the Software Improvement Group (SIG), once during the development phase in Q4/W2 and once at the end of the project in Q4/W8. The feedback given by SIG for the version uploaded in Q4/W2 should be considered for the final version. The product will be evaluated by the user in Q4/W2 and feedback will be given to steer further development. The evaluation of the final version in Q4/W8 is part of the final grade.

Final presentation

The main purpose of the final presentation is to demonstrate the final product as well as to show the highlights (key problems and their solutions) and reflect on the development process. Each presentation lasts for 20 minutes and needs to include a live demonstration of the system. The presentation is followed by 10 minutes of discussion.

Implementation

The implementation of the system will start in W6/Q3. The implementation follows an incremental and iterative software development process.

Planning

Each iteration needs to be planned according to the lightweight SCRUM planning documents described before. At the end of each iteration the plan is checked: which features have been finished, which tasks are still open. The results of the check are considered for the planning of the next iteration.

Test-driven

The implementation should be test-driven: first, implement a test for a feature and then start implementing the feature. Make use of an xUnit test framework.

Integrate the different parts of your implementation as soon and as much as possible. Use build environments such as ant or maven. Develop corresponding integration tests to check whether the integrated system works.

Always have a running version

The implementation follows the concept of "Always have a running system". After each iteration, there will be a tool demonstration session in which each project group gives a 10 minute demo of the most recent implemented feature(s) followed by a 20 minute discussion of the design, implementation, tests, and iteration planning.

Assessment

The following describes the assessment of the project groups. Basically, the grade is given to the group.

Calculation of the grade

The formula for determining the grade per project group is:

Overall score =

$$0.3 * \text{Context score} + 0.3 * \text{Product score} + 0.3 * \text{Process score} + 0.1 * \text{Presentation}$$

The context related grade is based on:

- Implemented software system
- Context seminar report

The product score consists of the software engineering and context related part. The software engineering related part is calculated by:

- Final architecture design

Source code of the final product as evaluated by SIG

The process score is calculated by:

- Product vision and planning
- Final report (with lightweight SCRUM plans)
- Peer reviews

The presentation score is obtained from the final presentation.

Projectvaardigheden and Informatievaardigheden

Note, that every student needs to pass (or have previously passed) these courses in order to obtain the grade on the context project.

Final product evaluation by SIG:

The source code and test code of the final product (if it has been implemented with PHP, Java, C, C++, or C#) will be evaluated by the Software Improvement Group (SIG). The code criteria used by SIG are:

- Readability (naming of classes, methods, attributes and variables)
- Structure (size of classes and methods)
- Complexity (complexity of methods, depth of inheritance trees)
- Documentation

In projects developed with other technologies for which an automated assessment by SIG is not possible, the Final Product will be assessed by the context teacher and assistants using the same criteria.

Plenary presentations

The best presentation from each context project will be selected and awarded with a presentation at the plenary session. The plenary session is open to all students who participated in the course and people involved. Exact date is to be announced.

Demo market

There will be a demo market organized in parallel with the plenary session. The location of the demo market will likely be the ground floor hallway of the faculty. Software can be presented with laptops and/or posters. Tables and poster boards will be available.

Organization

Each context project is performed in groups of 5 students. A maximum of 4 groups per context project is allowed. Students can choose their three preferred context projects; the final assignment will be done by the context project teachers. Students with a higher number of ECTS will be given the preference to do their selected context project.

In general, the process and product is the responsibility of the group (e.g., group determines the agenda of the meetings). Similar, request for feedback and support should come from the group.

Role of the teacher

- Provides a general description of the assignment:
 - goal
 - problem statement
 - prerequisites
 - references and contact persons
- Regular meetings with the TA (weekly)
- Make sure that the groups follow the software development process
- Supports and guides the external advisor
- Overall responsibility

Role of the teaching assistant (TA)

They represent the daily coach of the project groups:

- TA gives feedback on draft versions of the documents, ideas and solutions (but should not provide solutions).
- TA meets weekly with the students (initiative in meetings should come from the students, make sure that they prepare for each meeting).
- TA checks the progress and planning documents

Role of the external advisor

The external advisor should give user stories for setting up the product vision document as well as feedback on the development process and the final product. Feedback on the development process can be one or two times during the implementation phase. Feedback on the product can be one time during the implementation phase (during the demo session in Q4/W2) and at the presentation of the final product.

Context seminar

The context seminar runs in parallel to the context project and provides (additional) information on the problem context. We recommend to give the context seminar within the first five weeks. The information should help the students in defining the requirements and architecture of the system.

The result of the seminar is a report that accounts for the product score. The requirements and structure of the report depend on the problem context and will be determined by the teacher. The submission deadline is at the end of the third quarter.

Projectvaardigheden

The course *projectvaardigheden* is part of the context project and takes place in the first five weeks of Q3 (see Blackboard and the schedule). Note that this course needs to be passed by each student in order to obtain the final grade for the context project. Students who have already passed the course should inform the context project coordinators; they do not need to repeat it. For more information, see the schedule or contact Johan den Hartog <J.denHartog@tudelft.nl>, who organizes and gives this course.

Informatievaardigheden

This course is part of the context project and takes place in the first week of Q3 (see Blackboard and the schedule). Note that this course needs to be passed by each student in order to obtain the final grade for the context project. Students who have already passed the course should inform the context project coordinators; they do not need to repeat it. For more information on this course, see the rooster or contact Monique de Bont and/or Nicole Will <educationsupport-lib@tudelft.nl>, who organize this course.

End-terms

Based on the learning objectives the following end-terms are used:

- The student should be able to perform a 'state of the art' analysis on the topic addressed in the project assignment
- The student should be able to select and justify the appropriate programming techniques for the end product
- The student should be able to analyze relevant scientific literature and discuss the implications (written and verbally) in the context of the project
- The student understands the principals of the (lightweight) scrum approach for software engineering and should be able to manage a project accordingly
- The student should be able to present, concisely and critically, a software product from technical, functional and end-user perspective

Project coordinators

dr. ir. Dick de Ridder, d.deridder@tudelft.nl

dr. ir. Martin Pinzger, m.pinzger@tudelft.nl (until February 1, 2013)

dr. Andy Zaidman, a.e.zaidman@tudelft.nl (after February 1, 2013)