Fall 2018 Milestone 1: Use cases, High Level Requirements and Architecture

25.Oct.2018

Objective:

Based on the brief high level project description (posted on Moodle class site) the objective of Milestone 1 is to develop: a) initial high-level personas and use cases; b) from use cases develop high level functional requirements for the application, c) list high level architecture, frameworks and tools to be used (generally the same as in M0), and c) get the teamwork going. Note that these are only high-level requirements and specs with the idea to get early feedback and iterate before investing in developing more detailed specs and first prototype in Milestone 2. Future designs can deviate from Milestone 1 in the spirit of iterative SW design and development.

Initial input for your work is the final project high-level description (posted on Moodle), class slides on the topics as well as your SW and tool selection for M0. For use cases and functional specs, feel free to also use your own ideas, research similar applications that already exist, talk to your friends etc. Please consult class material on Use Cases, Requirements and Specs.

This is the first **team** milestone. The whole student team submits **one** milestone document for each Milestone 1 – 5, submission details are below.

You will discuss ongoing work on Milestone 1 during team session in each class and you can also send e-mail to instructors with questions.

Expected size of this document is about 6-10 pages, using font and spacing as in this document.

Content and structure for Milestone 1 document for review:

In the document for Milestone 1 you must cover all of the following subsections in exact order as below (have a separate numbered section for each) in one PDF file. We require that each subsection starts on the new page:

1. Executive Summary: Short description of the final product/application and its key advantages, novelty, value (up to 1 page). Make it as an executive summary – think of answering the question of why we should fund this project. We suggest you assign a name to your project for easier reference and good "marketing". This summary should be readable to a general manger/executive that is not a CS specialist and is used to explain and also to advertise/promote your project. Typical outline is: one paragraph on the motivation and importance of the application you are developing, followed by a paragraph on what your application will be doing and how it helps the

users (high level only, no jargon) and optionally what is unique and special in your design. At the end say in one paragraph something about your team (e.g. about your student startup team…).

2. Personas and Use Cases: Summarize key *personas* (categories of users) for your application – their general characteristics, goals, skills, pain points related to the application you are developing. About 1/3 of a page per persona – see class notes. (Note: in personas you stay general, in use cases you say how personas will use your app (at high level)). Then provide 4-5 main *use cases* (one paragraphs for each use case) - see class notes on more detailed format for requirements. Focus only on main use cases. Simple text format is OK and preferable – tell us a story about who and how the application is used. Focus on WHAT users do, their skill level, not on HOW is the SW implemented. NOTE: avoid specific on HOW functions will be done and text resembling user manual: this is supposed to guide the design of the future product and is NOT a description of how the product will work (you don't know that yet) – see class slides for details.

3.Data Definitions - define main terms, data structures and "items" or "*entities*" *at high or logical (not implementation) level* (e.g. name, meaning, usage, and NOT how the data is stored in memory) so it is easier to refer to them in the document. Focus on key terms (main data elements used in your app, types of users and their privileges etc.) specific for this application and not on general, well know terms. These terms and their names *must be used consistently* from then on in all documents, user interface, in naming SE components and database elements etc. In cases where you attach behavior and privileges to data items (e.g. user types) that also drives the design of the SW. In later milestones you will add more implementation details for each item. You will later expand this section with more details.

4.Initial list of functional requirements – see class notes. This refers to high level functions you plan to develop to the best of your knowledge at this point. Focus on WHAT and not HOW. Keep the user in mind. Develop these functions to be consistent with use cases and requirements above. Number each requirement with *unique numeric value* and use these numbers consistently from then on. For each functional requirement use 1-3 line description. At this stage no need to prioritize the requirements.

5.List of non-functional requirements (performance, expected load, security requirements, storage, availability, fault tolerance…) Number each. Note that mandatory high level non-functional specs are given in high level document, so for Milestone we recommend you simply copy them from high level document from iLearn. Please observe and adhere to these non-functional requirements in your

design and development from now on – you are not allow to change them unless you get permission.

6.Competitive analysis: Find 3-4 competitive products. Present competitors' features vs. your planned ones. First, create a table with key features of competitors vs. yours planed, only very high level, 5-6 entries max (as shown in the class). After the table, you must summarize in one paragraph what are the planned advantages or competitive relationship of your planned product to what is already available. In the table clearly mark your product, e.g. shade its column/data.

7.High-level system architecture Briefly provide itemized list of all main SW components such as frameworks, APIs, tools and systems to be used, supported browsers and deployment platform (SW and server) to be used.. This list is to be the list of approved tools and systems from M0. Any other external code/API/tool must be approved by instructors and you have to justify it.

8.Team: list student names, name of the team leader, names of front and back team lead and initial roles for each member

9. Checklist: for each item below you must answer with only one of the following: **DONE**; or **ON TRACK** (meaning it will be done on time, and no issues perceived); or **ISSUE** (you have some problems, and then define what is the problem with 1-3 lines)

- Team found a time slot to meet outside of the class
- Github master chosen
- Team decided and agreed together on using the listed SW tools and deployment server
- Team ready and able to use the chosen back and front end frameworks and those who need to learn and working on it
- Team lead ensured that all team members read the final M1 and agree/understand it before submission

Background reading:

- Document we posted about high-level vision of our application.
- Class material on requirements and specs
- Relevant existing applications and products.
- Info about allowed frameworks – class notes and posted on iLearn
- M0 document and documentation on SW tools and frameworks you plan to use
- Git and Github tutorials

Submission for Milestone 1 document for review – you must follow the instructions below PRECISELY:

Teams must collaborate in creating M1 document by having working M1 document on their team Github private repository (similar to managing code) so all team members and instructors can access it. Added advantage of doing it this way is that it builds teamwork and communication. We recommend having a folder for project documentation on team github where milestones and other similar files can be kept.

We strongly suggest the following M1 *collaborative approach (NOTE: creating a team document is similar to creating a code by the team of programmers)*:

- Team lead (or somebody assigned) creates a folder on Github team repo called "Milestones" where all milestone docs (including those final to be reviewed by instructors) will be placed
- Team leads assigns M1 editor,
- Team lead/M1 editor assign individual chapters to team members
- M1 editor collect chapters, edits/corrects then integrates them into a well formatted document (with same font and formats)
- M1 editor posts final candidate full document on team repo so that all team members read full document for one more review and any feedback
- M1 editor completes the final version as per feedback and submits as requested (make sure it is placed in correct directory, has the required file name etc.)
- Team lead submits M1 info for review as per submission instructions. Submission instructions (below) must be followed precisely and completely or grade penalty will be imposed

The whole student team submits one milestone document for Milestones 1, as follows: Team leads will send e-mail with a link (NOT the attached file) pointing directly to Milestone 1 Document (in PDF) to rainer.todtenhoefer@informatik.hs-fulda.de with e-mail subject line as specified below. Team leads have to ensure that both instructors have access to their Github team repository so they can open the link.

Submission e-mail subject line: MUST be " Fall 2018 Milestone1 Team N" in the subject line.

File name of the M1 document (to which the link is pointing) to MUST be: Fall 2018 Milestone1 Team *N.PDF* (*N* is your team number 01---15). (We use only PDF so I can send you feedback as yellow sticky notes).

M1 document format and structure

- **Title page** MUST include
  –"SW Engineering CSC648/848 Fall 2018"
  Project/application title and name (you can use the name you chose for your application)
  –Team number
  –Names of students (team lead first) with e-mail of team lead
  –"Milestone 1"
  –Date
  –History table (revisions) (Note: you will update this document based on instructors' feedback so this is important)

- **The rest of the document** has to contain sections as described above under "Content and structure for Milestone 1 document for review"

Team leads and M1 editors: make sure document is well formatted, reads well, is complete, and looks professional. This will be part of your portfolio and will influence the grade. Make sure all team members read final version and give comments before submission.

Instructor's feedback and creating final Milestone document for Final Project delivery

In the course of developing Milestone 1 you can ask instructor questions via e-mail and during team session in the last hour of the class. Upon submission of Milestone 1 you will get feedback from instructors by any of the following: e-mail, markings on your document and in class during team meetings. This feedback must be analyzed and taken into account by your team in order to revise your Milestone 1 and this must be used subsequently for the rest of the project. Please enter the revision summary in history table.

Instructors will comment from the standpoint of CEO, VP of Marketing (who translates customer and marketing requirements) and CTO (Architecture etc.). You may choose not to agree with the comments. This is OK as long as you justify this and are prepared to live with that design and deliver it. In some cases, instructors may insist on some features or decisions.

Upon getting instructors' feedback on your questions and submitted document, you need to revise your first draft, freeze it (meaning no more changes on this document even if future design changes) and use it as a basis for developing Milestone 2. This frozen document Milestone 1 will be submitted as part of final project delivery in Milestone 5. Do not start working on Milestone 2 before you get feedback on Milestone 1 and make sure all team members read frozen Milestone 1 document.

Future Milestone 2 functions and actual SW app may differ from what you proposed in Milestone 1, that is normal and in fact expected to happen in the spirit of iterative SW development. In that case there is no need to modify frozen M1 document.