Software-Project 2017

Review Document

Real-Time Mesh Utilities

Petros Simidyan	Blerta Hamzallari	Felix Griesau	Marco Klamke
Julius Lerm	Lars Debor	Simon Heinke	Sugandha Sachdeva

last change: May 1, 2017

Review Document

The review document is divided into two parts. It contains the results of planning as well as the outcomes of the design process. Results of planning include development model, cost and risk estimation, milestones and organization. The design process contains the outcomes of the current iteration, arrangements for the next iteration and a list of tools.

1 Result of Planning

A large part of the first phase of the project (i.e. scheduling and draft) is reflected on the functional specification document. The requirement analysis is registered, the objectives are declared, whereas the decisions and the product information is written down.

1.1 Software Development Model

This section contains information about the software model chosen, based on the requirements of the project. The principals of the group, client requirement and knowledge about the project play an important role in choosing the development model. Based on the latter, the development team decides its work flow.

Agile Development Model: SCRUM The group chose SCRUM because it is an iterative and incremental agile software development framework for managing product development. The duration of each sprint was set to two weeks. Each phase of the software development has two sprints.

Every sprint ends with a presentation by the relevant working group about the developments and progress during the sprint. The end of the respective phase of the project is marked by a working prototype and a presentation which includes a summary of the work done by the entire team.

Projects specific adaptation to the model: Every person in the team has multiple roles. All group members work on both the document and the code.

1.1.1 Software Development Specific Content

Since the group decided for the agile development project, the milestones need to be stated and agreed upon by the team. Milestones are the aim or the expected output of each development phase. They help the team to specify what features should be completed by which deadline.

1.2 Effort Estimate

The main purpose in the effort estimation section is the categorization of the different parts of the project regarding their complexity and effort criteria. (see figure 1)

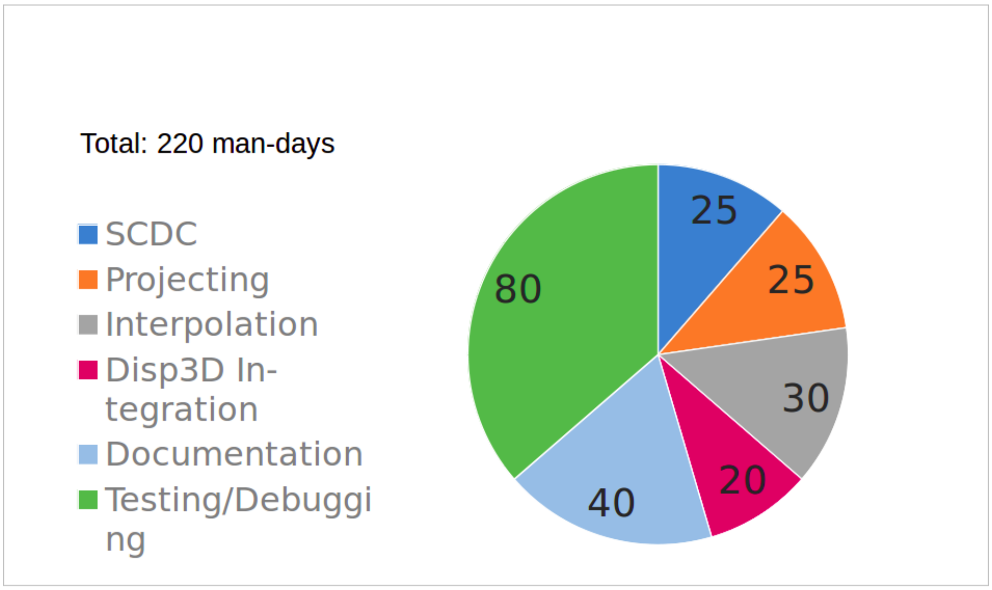


Figure 1: Effort estimate

1.3 Risk Estimate

In this section, the probabilities of the different risks involved in the project are listed. This will help the team to determine what aspects of the implementation should get a higher priority.

RE1: Communication problems in the team

RE2: Coverage is too extensive

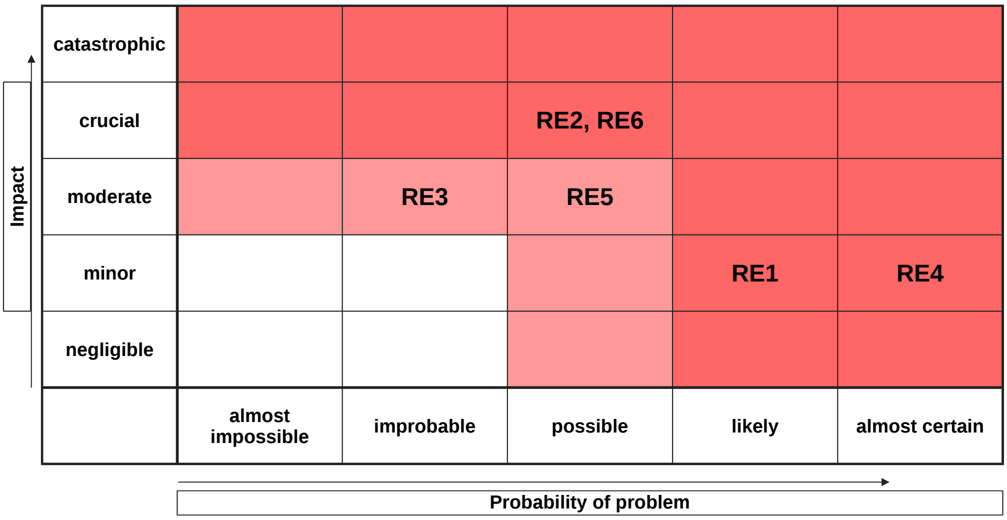
RE3: Framework does not provide the needed functionality

RE4: The resource bottleneck derived from team members absence

RE5: Change of the requirements due to miscommunication with the product owner

RE6: Hidden complexity

RE7: An acceptable computation time takes a lot more effort to be achieved, than expected



catastrophic					
crucial			RE2, RE6		
moderate		RE3	RE5		
minor				RE1	RE4
negligible					
	almost impossible	improbable	possible	likely	almost certain

Impact
Probability of problem

Figure 2: Risk estimate

1.4 Milestones

First Phase Milestones: functional specification, preliminary design, review document, first prototype, presentation

Second Phase Milestone: SCDC, projecting, interpolation, integration Disp3D, review document, presentation, detailed design

Third Phase Milenstone: Portation to MNE Scan, SCDC (tested and operating), projecting (tested and operating), interpolation (tested and operating), review document, presentation

1.5 Organization

This section concerns to the rules, agreements and the partitioning regarding the teamwork in the project, so the work itself will be efficient and organized.

1.5.1 Ways of Communication

Telegram: Used for quick and direct team communication so that possible misunderstandings will be solved in no time.

E-mail distribution list: Used for scheduling the team meetings and communications with the extended team, including the product owners.

Team meetings: Used for the review and direct discussion of the encountered problems.

Skype: Used in the cases of the absence of a team member.

Jira: Used for scheduling tasks and keeping track of the progress done by each member of the team.

Dropbox: Used for exchanging documents and file sharing.

1.5.2 Additional Agreements

- Internal team meetings (without product owners): Tuesdays and Thursdays at 19:00
- External team meeting (with the product owners): Wednesdays at 17:00
- Meeting of subgroups : upon consultation and demand

1.5.3 Role Assignment in SCRUM

Product Owner: Thomas Jochmann, Lorenz Esch

Scrum Master: Simon Heinke

Development team: Blerta Hamzallari, Felix Griesau, Julius Lerm, Lars Debor, Marco Klamke, Simon Heinke, Sugandha Sachdeva, Petros Simidyan

Client, User: Participants of the MNE CPP project of Boston Children's Hospital

1.5.4 Role Assignment Organization

Advisor: Thomas Jochmann, Lorenz Esch

Team leader: Simon Heinke

Build master: Lars Debor

Version management : Felix Griesau

2 Results of Design

The results of design can be found in the preliminary design document. The connections between different packages, components and classes are explained and visualized using UML-diagrams.

2.1 Tools

The used tools are programs which facilitate the different aspects of the organization and development.

2.1.1 Organization Tools

Code versioning: The source code will be managed via *Git* to ensure a convenient exchange and tracking of code.

LaTeX: *LaTeX* is used for editing review documents and documentation.

Doxygen: This program will automatically document source code comments.

Visual Paradigm: Used for creating UML-diagrams and other aspects of visual documentation.

2.1.2 Developing Tools

Development environment: Since it is already established within MNE-CPP, *QtCreator* will be used for code editing and compiling.

Program language: *C++* will be used hence it is the main language of MNE-CPP

Operating systems: The software will be developed on Linux and Microsoft Windows.

Libraries: Besides standard libraries from the C++ 11 standard, the MNE-CPP framework is based on Eigen, OpenGL and Qt.

2.2 Results of first iteration design:

The results of the first iteration correlate with the respective milestone.

Functional specification: The requirements specification is transferred into the corresponding functional specification.

Preliminary design: The preliminary design includes a first overview of the main features and structure (see preliminary design document).

Implementation: A first executable prototype is created.

Planning: Milestones for the next iteration are declared.

2.3 Tasks for next iteration

Refining the preliminary design : The preliminary design is extended with diagrams and more detailed descriptions. This will be done according to milestones progress.

Further implementation: The mentioned features will be implemented and tested.