Project Plan

SQUID

Helsinki 8th February 2005

Software Engineering Project

UNIVERSITY OF HELSINKI
Department of Computer Science

Course

581260 Software Engineering Project (6 cr)

Project Group

Mikko Jormalainen
Samuli Kaipiainen
Aki Korpua
Esko Luontola
Aki Sysmäläinen

Client

Lauri J. Pesonen

Project Masters

Juha Taina
Jenni Valorinta

Homepage

`http://www.cs.helsinki.fi/group/squid/`

Change Log

Version	Date	Modifications
0.1	25.1.2005	First version
0.2	28.1.2005	Process model and schedule modified, other additions
0.3	3.2.2005	Schedule, risks, converted to latex, translated to English
0.4	7.2.2005	Suggested changes from last meeting
1.0	8.2.2005	Approved with modifications

Contents

1 Introduction

The Geophysics Department of the University of Helsinki uses a SQUID magnetometer for measuring the magnetism of rocks and meteorites. There is already a program for using the machine, but its usability could be better. The source code of the existing program is available.

The goal of this project is to improve the existing program by making a better user interface for it. A secondary goal is to link the measurements produced by the program to desired after-processing programs.

The project will take place from 25.1.2005 to 6.5.2005.

2 Organization

The people related to this project are shown in Figure 1.

Name	Role	E-Mail	Phone
Mikko Jormalainen	Project Team	mtjormal@cc.helsinki.fi	040 545 6249
Samuli Kaipiainen	Project Team	samuli.kaipiainen@cs.helsinki.fi	050 357 0359
Aki Korpua	Project Team	aki.korpua@cs.helsinki.fi	044 339 5375
Esko Luontola	Project Team	esko.luontola@cs.helsinki.fi	041 507 1474
Aki Sysmäläinen	Project Team	aki.sysmalainen@helsinki.fi	0400 603 416
Lauri J. Pesonen	Client	lauri.pesonen@helsinki.fi	
Tomas Kohout	Client	tomas.kohout@helsinki.fi	
Fabio Donadini	Client	fabio.donadini@helsinki.fi	
Juha Taina	Course manager	taina@cs.helsinki.fi	09 191 51311
Jenni Valorinta	Instructor	valorint@cs.helsinki.fi	09 191 51165

Figure 1: The people who are part of this project

2.1 Responsibilities

The responsibilities of the project team members are described here. The assignments are shown in Figure 2.

Project Manager
Is responsible for leading the project team. Writes the project plan and takes care that the project progresses according to it.

Secretary
The secretary changes every week. Writes the meeting records and forwards them to the team.

CVS Manager
Sets up the CVS system, takes care of it and teaches the team members.

Responsibility	Person
Project Manager	Esko Luontola
Secretary	Mikko Jormalainen, Samuli Kaipiainen, Aki Korpua, Aki Sysmäläinen
CVS Manager	Aki Korpua
HTML Manager	Esko Luontola
Measurement Manager	Aki Sysmäläinen
Project Plan	Esko Luontola
Requirements Document	Aki Sysmäläinen, Aki Korpua
Design Document	Mikko Jormalainen, Samuli Kaipiainen
Testing Plan	Aki Korpua, Mikko Jormalainen
Testing Report	Aki Korpua, Mikko Jormalainen
User Manual	Aki Sysmäläinen, Samuli Kaipiainen
Realization Document	Aki Sysmäläinen, Samuli Kaipiainen
Final Report	Esko Luontola

Figure 2: Assigned responsibilities

HTML Manager
Updates the web site and adds the published documents there.

Measurement Manager
Takes the measurements that are gathered from OHTU projects and submits them to the course manager.

3 General Description of the Product

The SQUID magnetometer is used for measuring the magnetism of rocks and meteorites. The machine consists of three main components: measurement unit, demagnetizing unit and sample holder. The computer guides their operation through COM ports. The existing source code will be used to control the machine on a low level, and on top of it will be built a new user interface.

The program must be able to control the machine, gather measurements from the samples, show them to the user and save the results on disk for later use. The program will be used on a modern PC running Windows. The old program has been written in Visual C. The new program will be written either in C and Java, or in C++.

The name of the program will be Ikayaki (dried, grilled squid - a Japanese seafood).

4 Estimated Size

The size of the old source code is about 12,000 lines of code. Parts of the old source code can be reused, but on the other hand the new program will have more features than the old

one. Taking these into account, it can be estimated that we will need to write about 10,000 lines of code (+-30%). A better estimation will be made when the program requirements are known better.

Previous experiences from OHTU projects show that in average people do write 300-350 lines of code per person per month. The low numbers are mainly because of the documentation and testing that require much time. So statistically our team should be able to produce about 5,000 lines of code. We can conclude that it will *not* be possible to complete this program in full detail in this time. Thus we will try to make the requirements analysis and user interface designing as well as possible, so that other teams could continue from where we were left.

5 Workflow

The process of developing this program is described here. There are two concurrent processes: one for finding out the program requirements and building the user interface, and one for finding out how the source code works and building an application interface for using the SQUID.

5.1 Main Process: GUI and producing the program

The process for creating the new user interface will follow a linear waterfall model. The output of the previous phase will be the input of the next one.

5.1.1 Definition

Requirements analysis will be done with the client and the program requirements will be listed. The client will guide the project team in using the existing program and magnetometer. The results will be requirements document and user interface prototype. The client will accept the documents before proceeding.

5.1.2 Design

The user interface and structure of the program will be designed. The expertise of the client will be needed during this phase. Designing will be made so accurately that producing the program will be as simple as possible. The results will be design document and testing plan. The client will accept the documents before proceeding.

5.1.3 Production

The program will be written according to the design document. Each programmer will take care of the unit testing of his own code. The result will be program code.

5.1.4 Testing

The program will be tested according to the test plan and the found errors will be corrected. Uncorrected errors will be documented. User manual will be written. The results will be testing report, user manual and realization document.

5.1.5 Installing

The program will be given to the client and the project team will install it in the laboratory. Possibly an installer will be created for wider distribution. Documentations will be tweaked and final report written.

5.2 Sub Process: analyzing the old source code

Analyzing the old source code, designing a suitable interface for using the SQUID and building it will be executed concurrently with the main process. The interface needs to be defined before the main process will need the information in its design phase.

The sub process will utilize 1-2 persons.

5.2.1 Design

Getting to know the old source code and finding out how the program works. An interface for using the magnetometer will be designed. The result will be interface design document and testing plan, which will be attached to the actual design document and testing plan.

5.2.2 Production

The interface will be built according to the design plan.

5.2.3 Testing

The interface will be tested with real equipment according to the test plan. If the interface is completed much before the user interface is ready, it might be necessary to build a simple user interface for testing purposes only. Otherwise the interface will be tested with the rest of the program.

5.3 Schedule

The length of the project is about 14 weeks, starting 25.1.2005 and ending latest 6.5.2005. The lengths of the phases are shown in Figure 3. They are also described graphically using a GANTT diagram in Figure 4.

Task	Start	End	Days	Results
Project Plan	25.1.2005	15.2.2005	21	Project Plan
Definition	25.1.2005	22.2.2005	28	Requirements Document, Prototype
(Main) Design	22.2.2005	15.3.2005	21	Design Document, Testing Plan
(Sub) Design	25.1.2005	22.2.2005	28	(Design Document, Testing Plan)
(Main) Production	15.3.2005	29.3.2005	14	Program Code
(Sub) Production	22.2.2005	1.3.2005	7	Program Code
(Main) Testing	29.3.2005	26.4.2005	28	Testing Report, User Manual, Realization Document
(Sub) Testing	1.3.2005	22.3.2005	21	(Testing Report)
Installing	26.4.2005	3.5.2005	7	Final Report

Figure 3: Project schedule

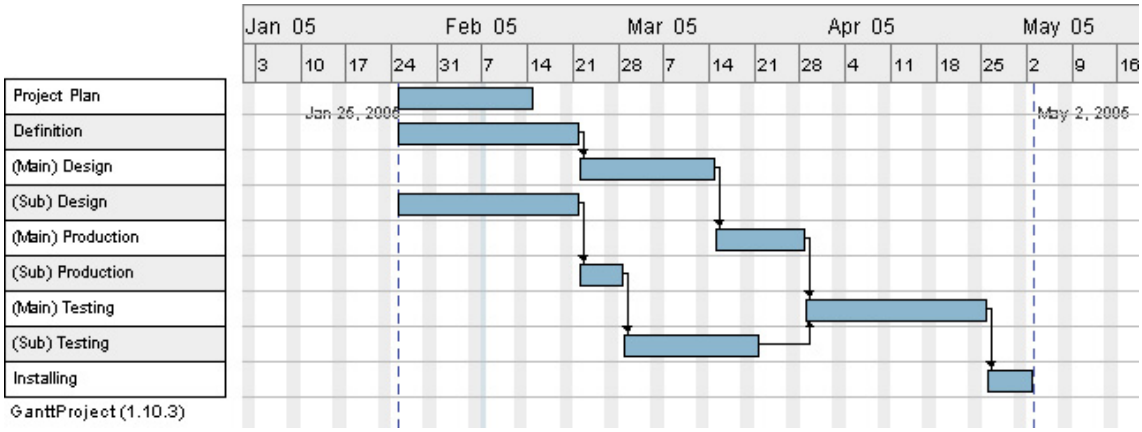

Figure 4: Project schedule as a GANTT diagram

6 Conventions

Meetings will be held every Tuesday at 10-12 and Thursday at 10-12 in Exactum class A219. The secretary will write a record and publish it as soon as possible after the meeting.

Every meeting should have a pre-made list of the items that need to be discussed in that meeting. If somebody won't be able to make it to a meeting, he should report it beforehand. The items should be discussed in an orderly manner and an item needs to be completed before moving to the next one. Documents that will be handled in a meeting must be published the day before the meeting before 18:00.

Work sessions will be held on Wednesdays at 14-18 when necessary. The team will meet in the Exactum lobby by default.

Communication outside the meetings will be through e-mail and the mailing list ohtuk05-squid-list@cs.helsinki.fi. All e-mail messages should be prefixed with (squid). The project has an IRC channel #squid-project at IRCnet.

The material produced in the project will be managed with CVS. The CVSROOT is

/home/group/squid/.cvsroot

Every team member should write what he is doing currently to a text file so that it could be viewed from the web site:
/group/home/squid/public_html/username/status.txt

The team uses the locker 1 in room A307.

7 Documentation

During the project the following documents will be published. They will be made with LaTeX and published in PDF format. Documents will be stored in the CVS during the project.

- Project plan
- Requirements Document
- Design Document
- Testing Plan
- User Manual
- Testing Report
- Final Report
- Realization Document
- Meeting Records (in Finnish)

8 Risk Analysis

8.1 Team

Risk: Communication in a foreign language causes problems.
Propability: Low
Seriousness: Medium
Countermeasures: Those who known English better, will be used to communicate with the client.

Risk: A team member gets sick.
Propability: Medium
Seriousness: Medium
Countermeasures: The rest of the team should be informed as soon as possible, so that others can share the work when necessary.

Risk: A team member quits the project.
Propability: Low
Seriousness: High
Countermeasures: Maintain a happy and encouraging atmosphere.

8.2 Project Management

Risk: The estimations (size of work, skills) are incorrect.
Propability: High
Seriousness: Medium
Countermeasures: Keep and eye on the project plan and update it as necessary.

Risk: The project will not stay in schedule.
Propability: High
Seriousness: High
Countermeasures: Look at the schedule at the end of every week to notice problems as soon as possible. Adjust the schedule when necessary. Drop secondary features off the program (do not neglect testing!).

Risk: The project will not be completed in time.
Propability: Medium
Seriousness: High
Countermeasures: If the program appears too big for us to complete it, produce only the core elements of it, but do them well. It is possible that some other team will then continue from where we were left.

Risk: Communication is not adequate.
Propability: Medium
Seriousness: Medium
Countermeasures: Arrange regular meetings between the team members. Follow closely what everybody is doing at the moment and what the results are.

Risk: Participants are not interested in the project.
Propability: Low
Seriousness: High
Countermeasures: Keep everybody informed about the status of the project to keep their interest up. Show the current results for the client when there is something to show. Arrange interesting work for the team members.

Risk: Some personal interest takes the time of a team member.
Propability: Medium
Seriousness: Medium
Countermeasures: Follow closely what everybody is doing. If somebody does not show progress, find out what is wrong. Tell others if you know that you will not have enough time.

8.3 Technology

Risk: The old source code and the operation of the machine is not being understood well enough, and there will be problems in the reuse of the code.
Propability: Medium
Seriousness: High
Countermeasures: Allocate more people to have a look at the source code. Discard the old source and start from a scratch, if the inclusion of legacy code proves to be too risky.

Risk: The SQUID gets broken.
Propability: Low
Seriousness: High
Countermeasures: Test the code well before trying it with the real machine.

Risk: If some need to learn a new programming language, their code will have more bugs and they will write it slower.
Propability: High
Seriousness: Medium
Countermeasures: Utilize the existing knowledge of the individuals as much as possible. Try coding in pairs. Do thorough testing. Make one or two people learn the new stuff well, so that others won't need to spend as much time on it.

Risk: If some need to learn using new tools, they will be slow in using them.
Propability: Medium
Seriousness: Low
Countermeasures: Take the learning curve into consideration and adjust the schedule. Make one or two people learn the new stuff well, so that others won't need to spend as much time on it.

Risk: A person is not skilled enough in the task that has been assigned to him.
Propability: Low
Seriousness: Low
Countermeasures: Assign that person to do some other work where he is better. Try to utilize the special skills of the team members as much as possible, especially in case of a difficult task.

8.4 Client

Risk: The project team will not understand how the magnetometer is being used.
Propability: Medium
Seriousness: High
Countermeasures: Check all decisions concerning the program design with the client. Build an accurate prototype of the system and make the client accept it before writing code.

Risk: The client you tried to contact can not be reached.
Propability: Low

Seriousness: Medium
Countermeasures: Find out beforehand if the client knows when he will be unavailable. Try to do other things that do not require the client while waiting. They to become independent of the client after the beginning of the project.

Risk: The client will not be satisfied with the finished product or the product will be useless.
Propability: Medium
Seriousness: High
Countermeasures: Build prototypes and write documents that will be accepted by the client. Test the program thoroughly.

Risk: The client does not understand the prosess of making software.
Propability: High
Seriousness: Low
Countermeasures: Explain to the client what we are doing. Remind the client not to expect too much.