

# **Requirements document 1.0**

SQUID

Helsinki 1st March 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  
Fabio Donadini  
Tomas Kohout

## Project Masters

Juha Taina  
Jenni Valorinta

## Homepage

<http://www.cs.helsinki.fi/group/squid/>

## Change Log

Version	Date	Modifications
0.1	9.2.2005	First version (Aki Sysmäläinen)
0.2	18.2.2005	Use case list (Samuli Kaipiainen)
0.3	21.2.2005	Some use cases expanded (Samuli Kaipiainen)
0.4	22.2.2005	User requirements (Mikko Jormalainen)
0.5	23.2.2005	More use cases expanded (Samuli Kaipiainen)
0.6	25.2.2005	All use cases expanded (Samuli Kaipiainen)
0.7	26.2.2005	More user requirements, architecture (Esko Luontola) Use case diagram (Samuli Kaipiainen)
0.8	27.2.2005	Use case -> requirements -references (Samuli Kaipiainen) UI protos, verified the use cases (Esko Luontola)
0.9	28.2.2005	Validation (Mikko Jormalainen) Requirement -> uce cases -references (Mikko Jormalainen) Requirement setters (Mikko Jormalainen) Overview (Aki Sysmäläinen) Requirement summary table (Samuli Kaipiainen) Description of UI design methods (Samuli Kaipiainen) Goal-derived use cases (Esko Luontola)
1.0	1.3.2005	Random corrections

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Overview</b>	<b>1</b>
<b>3</b>	<b>Use cases</b>	<b>2</b>
3.1	Measuring . . . . .	2
3.2	Exporting . . . . .	6
3.3	Printing . . . . .	8
3.4	Projects . . . . .	8
3.5	Sequences . . . . .	10
<b>4</b>	<b>Requirements</b>	<b>14</b>
4.1	Functional requirements . . . . .	15
4.1.1	Magnetometer . . . . .	15
4.1.2	Project files . . . . .	16
4.1.3	Measurements . . . . .	17
4.1.4	Others . . . . .	18
4.2	Quality requirements . . . . .	18
4.3	Environment . . . . .	19
4.4	Maintainability . . . . .	20
<b>5</b>	<b>User interface</b>	<b>20</b>
5.1	Goal-derived use cases . . . . .	20
5.1.1	Use case 1: Erkki studies the movement of Antarctis . . . . .	20
5.1.2	Use case 2: Anomaly in magnetic field at year 500 . . . . .	21
5.1.3	Use case 3: Tomas studies meteor magnetism . . . . .	22
<b>6</b>	<b>Architecture overview</b>	<b>23</b>
<b>7</b>	<b>External interfaces</b>	<b>24</b>
7.1	Existing program . . . . .	24
7.2	Hardware control protocols . . . . .	24
<b>8</b>	<b>Validation</b>	<b>24</b>

## **Appendices**

**1 Use case 1: Erkki studies the movement of Antarctica**

**2 Use case 2: Anomaly in magnetic field at year 500**

**3 Use case 3: Tomas studies meteor magnetism**

**4 Automated Sample Handler System Protocol**

**5 Automatic Sample Degaussing System Protocol**

**6 Superconducting Rock Magnetometer Protocol**

# 1 Introduction

This document describes requirements for a SQUID magnetometer program that will be designed and implemented as a software engineering student project at University of Helsinki at the Computer Science Department. The clients is Lauri Pesonen with his assistants Fabio Donadini and Tomas Kohout from the Department of Geophysics.

This document serves as a contract between client and the project group about the implemented functions of the program. Particularly the use cases and requirements sections of this document describes those functions.

# 2 Overview

Department of Geophysics uses a magnetometer that works in SQUID (superconducting quantum interference device) principle to measure magnetization of minerals and rocks. Magnetometer is controlled by a computer software that controls and reads electric components that control the device itself.

The use of the present magnetometer software is complicated and unnecessarily burdens the users work memory with complicated work phases and divergent operation in many use cases. The work flow now requires the use of various programs in different phases of the measurement, which is quite complicated and slow. The functions of those programs could be combined to one. The present programs are also hard to learn and teach which limits the group of users.

User of the device must know clearly what its state is and which operation is being performed at the moment. The device is delicate and expensive, so a malfunction of it must be noticed early to prevent any damage. For example faulty state of demagnetizer can damage the device if it's not shut down manually in time.

The device and program are used to carry out long measurement sequences, involving several stages. It's important that these sequences can be performed flexibly and fast. User must be able to follow the measuring procedure and to stop it and modify the sequence if there are unexpected changes in the results.

Further processing and analyzing of the results occurs in various other programs which use a standard file format. So the new program has to be able to export these formats.

### 3 Use cases

This section describes planned use cases for the program. Use cases are derived from user interface prototype and requirements. All use cases are performed by any ordinary user and in program main screen. A simplified use case diagram is presented in Figure 1.

Use cases are divided to different sections, grouping logically similar use cases together.

Use case format:

#### **UC0: Use case identifier and title**

[POSSIBLE TEMPORARY NOTE HERE FOR PROJECT GROUP; SHALL BE REMOVED FROM FINAL VERSION.]

<b>Scenario a</b>	First scenario for doing the use case.
<b>Scenario b</b>	Second scenario for doing the use case. ...
<b>Precondition</b>	Preconditions for use case.
<b>Postcondition</b>	Postconditions for use case.
<b>Error condition</b>	Error handling, mainly if anything special needs to be done.
<b>Goal-derived</b>	Goal-derived use case(s) in which this use case occurs (if any); see section 5.1.
<b>Requirements</b>	Requirement(s) from which this use case derives; see section 4.

#### 3.1 Measuring

*As in any and all measuring action with the squid.*

#### **UC1: Single step measuring without demagnetization**

<b>Scenario a</b>	Enter as next AF demagnetization step "0" or empty (default for new projects), meaning no demagnetization, and click "Single step".
<b>Precondition</b>	AF project open, sample in sample holder.
<b>Postcondition</b>	Sample measured, results on screen and appended to current project.
<b>Error condition</b>	The program shall let the user know if something went wrong.
<b>Goal-derived</b>	5.1.3, first few steps.
<b>Requirements</b>	R1, R6, R10, R14, R17, R18

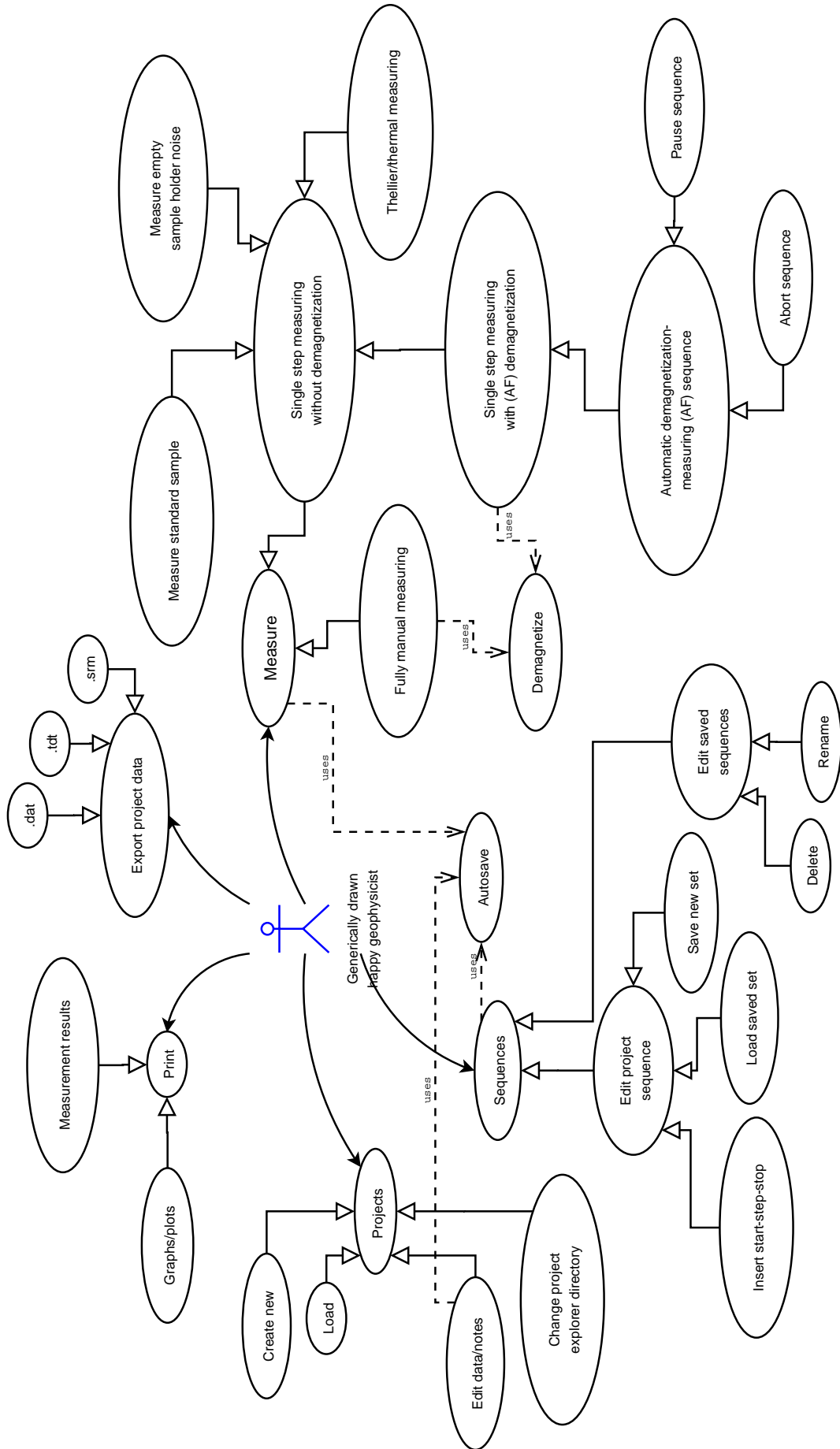


Figure 1: Use case diagram

**UC2: Single step measuring with (AF) demagnetization**

<b>Scenario a</b>	Enter as next AF demag step anything greater than zero, and click "Single step".
<b>Precondition</b>	AF project open, sample in sample holder.
<b>Postcondition</b>	Sample demagnetized (possibly ruined) and measured, results on screen and appended to current project.
<b>Error condition</b>	The program shall let the user know if something went wrong, and, should the demagnetization field not be coming down, warn user with an alarm sound.
<b>Goal-derived</b>	-
<b>Requirements</b>	R1, R4, R6, R10, R14, R17, R18

**UC3: Automatic demagnetization-measuring sequence (AF sequence)**

<b>Scenario a</b>	Enter the AF sequence (see section 3.5 for ways to enter it) and click "Measure".
<b>Precondition</b>	AF project open, sample in sample holder.
<b>Postcondition</b>	Sample demagnetized according to entered AF sequence (possibly ruined) and measured after each demagnetization, results on screen and appended to current project.
<b>Error condition</b>	The program shall let the user know if something went wrong, and, should the demagnetization field not be coming down, warn user with an alarm sound.
<b>Goal-derived</b>	5.1.1 whole sequence, 5.1.3 sequence after few single steps at the beginning.
<b>Requirements</b>	R1, R4, R6, R10, R13, R17, R18

**UC4: Pause automatic measuring sequence**

<b>Scenario a</b>	While measure sequence is running, click "Pause".
<b>Precondition</b>	Ongoing measuring sequence (UC3).
<b>Postcondition</b>	Measure sequence halts after current step is done, results on screen.
<b>Error condition</b>	Program tells if sequence can't be paused (and something has gone terribly wrong).
<b>Goal-derived</b>	5.1.3, Tomas pauses the sequence as the meteor demagnetizes too rapidly.
<b>Requirements</b>	R1, R15

**UC5: Abort automatic measuring sequence**

<b>Scenario a</b>	While measure sequence is running or paused, click "Stop now!".
<b>Precondition</b>	Ongoing measuring sequence (UC3).
<b>Postcondition</b>	Measure sequence halts immediately [AND PROGRAM ENTERS "FULLY MANUAL" MODE?]
<b>Error condition</b>	Program tells if sequence can't be aborted (and something has gone terribly wrong).
<b>Goal-derived</b>	-
<b>Requirements</b>	R1, R16



**UC6: Thellier measuring**

<b>Scenario a</b>	Click "Single step".
<b>Precondition</b>	TH project open, sample in sample holder.
<b>Postcondition</b>	Sample measured, results on screen and appended to current project.
<b>Error condition</b>	As usual.
<b>Goal-derived</b>	5.1.2, temperatures already entered, one step at 380 °C.
<b>Requirements</b>	R1, R6, R10, R14, R17, R18

**UC7: Thermal measuring**

[EXACTLY THE SAME AS THELLIER?]

<b>Scenario a</b>	Click "Single step". (Temperature can be entered later, as it won't affect measuring.)
<b>Precondition</b>	TH project open, sample in sample holder.
<b>Postcondition</b>	Sample measured, results on screen and appended to current project.
<b>Error condition</b>	As usual.
<b>Goal-derived</b>	-
<b>Requirements</b>	R1, R6, R10, R14, R17, R18

**UC8: Measure magnetometer ground noise**

[2005-02-23 NOT IN CURRENT UI PROTO, NOR PLANNED FOR IMPLEMENTATION.]

<b>Scenario a</b>	Click "Ground noise" and "Calibrate".
<b>Precondition</b>	No ongoing measurements.
<b>Postcondition</b>	Ground noise measured, results on screen and appended to "Ground noise" project.
<b>Error condition</b>	-
<b>Goal-derived</b>	-
<b>Requirements</b>	R1, R3, R6, R17, R18

**UC9: Measure empty sample holder noise**

<b>Scenario a</b>	Click "Holder noise" and "Calibrate".
<b>Precondition</b>	No ongoing measurements, empty sample holder.
<b>Postcondition</b>	Holder noise measured, results on screen and appended to "Holder noise" project.
<b>Error condition</b>	-
<b>Goal-derived</b>	5.1.1, calibration at the beginning.
<b>Requirements</b>	R1, R3, R6, R17, R18

**UC10: Measure standard sample**

<b>Scenario a</b>	Click "Standard sample" and "Calibrate".
<b>Precondition</b>	No ongoing measurements, standard sample in sample holder.
<b>Postcondition</b>	Standard sample measured, results on screen and appended to "Standard sample" project.
<b>Error condition</b>	-
<b>Goal-derived</b>	-
<b>Requirements</b>	R1, R3, R6, R17, R18

**UC11: Fully manual measuring**

[2005-02-25 STILL NOT CLEAR HOW FULLY MANUAL IS SUPPOSED TO WORK; TOMAS WILL BE BACK ON TUESDAY 1.3. AND MIGHT TELL US.]

<b>Scenario a</b>	Click any of the manual control components.
<b>Precondition</b>	Manual mode enabled.
<b>Postcondition</b>	Manual action done, result on screen [AND APPENDED TO CURRENT PROJECT? DRAW GRAPHPLOTS?].
<b>Error condition</b>	-
<b>Goal-derived</b>	-
<b>Requirements</b>	R1, R2, R6, R10, R17, R18

**UC12: Enable manual mode**

[2005-03-01 DELETED USE CASE, STILL HERE TO REMIND IT EXISTED, AND, TO KEEP REFERENCES WORKING :)]

<b>Scenario a</b>	Click "Manual" checkbox above the manual control components.
<b>Precondition</b>	No ongoing measurement.
<b>Postcondition</b>	Manual mode enabled.
<b>Error condition</b>	-
<b>Goal-derived</b>	-
<b>Requirements</b>	R2

**3.2 Exporting**

*As in exporting measurement results to another file format for other programs to use.*

**UC13: Export project data into .dat file**

<b>Scenario a</b>	In project explorer file list, right-click on desired project file, click "Export .dat in current directory".
<b>Scenario b</b>	In project explorer file list, right-click on desired project file, click "Export .dat to disk drive A:".
<b>Scenario c</b>	In project explorer file list, right-click on desired project file, click "Export .dat...", choose directory and filename to export.
<b>Scenario d</b>	In project explorer file list, select multiple files with shift-click and ctrl-click, and make any of above actions.
<b>Precondition</b>	At least 1 project file in current (selected) directory.
<b>Postcondition</b>	Project data exported to .dat file.
<b>Error condition</b>	Notify if file error occurs (such as no disk in A: drive).
<b>Goal-derived</b>	-
<b>Requirements</b>	R11

**UC14: Export (thellier) project data into .tdt file**

<b>Scenario a</b>	In project explorer file list, right-click on desired project file, click "Export .tdt in current directory".
<b>Scenario b</b>	In project explorer file list, right-click on desired project file, click "Export .tdt to disk drive A:".
<b>Scenario c</b>	In project explorer file list, right-click on desired project file, click "Export .tdt...", choose directory and filename to export.
<b>Scenario d</b>	In project explorer file list, select multiple files with shift-click and ctrl-click, and make any of above actions.
<b>Precondition</b>	At least 1 project file in current (selected) directory.
<b>Postcondition</b>	Project data exported to .tdt file.
<b>Error condition</b>	Notify if file error occurs (such as no disk in A: drive).
<b>Goal-derived</b>	-
<b>Requirements</b>	R11

**UC15: Export single measurement details into .srm file**

<b>Scenario a</b>	In measurement result table, right-click on desired measurement line(s), click "Export .srm in current directory".
<b>Scenario b</b>	In measurement result table, right-click on desired measurement line(s), click "Export .srm to disk drive A:".
<b>Scenario c</b>	In measurement result table, right-click on desired measurement line(s), click "Export...", choose directory and filename to export.
<b>Scenario d</b>	In measurement result table, select multiple lines with shift-click and ctrl-click, and make any of above actions.
<b>Precondition</b>	At least 1 measurement result in current project.
<b>Postcondition</b>	Measurement details exported to .srm file.
<b>Error condition</b>	Notify if file error occurs (such as no disk in A: drive).
<b>Goal-derived</b>	-
<b>Requirements</b>	R11

### 3.3 Printing

*As in printing measurement results as text or graphs.*

[2005-02-25 NOT IN CURRENT UI PROTOTYPE, IMPLEMENTATION PRIORITY LOW.]

#### UC16: Print measurement results

<b>Scenario a</b>	Click "Print...", "Measurement results".
<b>Precondition</b>	Open project.
<b>Postcondition</b>	Measurement results printed via [Java] standard printing window.
<b>Error condition</b>	Let know if printing error occurs.
<b>Goal-derived</b>	-
<b>Requirements</b>	R17

#### UC17: Print graph sheet (with 7 different graphs; described elsewhere)

<b>Scenario a</b>	Click "Print...", "Graph sheet".
<b>Precondition</b>	Open project.
<b>Postcondition</b>	Measurement results printed via [Java] standard printing window.
<b>Error condition</b>	Let know if printing error occurs.
<b>Goal-derived</b>	-
<b>Requirements</b>	R18

### 3.4 Projects

*As in project files, which store all measurement results.*

#### UC18: Automatically save all measurement cycles in project file

<b>Scenario a</b>	Make any measurement action (see section 3.1 for those).
<b>Precondition</b>	Project file open.
<b>Postcondition</b>	After measurement (step) is done, new results appended to project file.
<b>Error condition</b>	-
<b>Goal-derived</b>	5.1.1, 5.1.2, 5.1.3.
<b>Requirements</b>	R6, R10

#### UC19: Create new project

<b>Scenario a</b>	Click the empty line (which is always visible) below filenames in project explorer, enter new project name, choose AF/TH (thellier) project, click "Create new" or press enter.
<b>Precondition</b>	Project explorer in desired directory (UC21).
<b>Postcondition</b>	New project created and selected.
<b>Error condition</b>	-
<b>Goal-derived</b>	5.1.1 after calibration, 5.1.3 at the beginning.
<b>Requirements</b>	R5

**UC20: Load project**

<b>Scenario a</b>	Click any filename in project explorer.
<b>Precondition</b>	Project explorer in desired directory (UC21). A measurement can be running at the same time (it will be highlighted in the project explorer).
<b>Postcondition</b>	Existing project loaded and selected.
<b>Error condition</b>	-
<b>Goal-derived</b>	5.1.2 all samples already as project files.
<b>Requirements</b>	R7, R10

**UC21: Change project explorer directory**

<b>Scenario a</b>	Click into current directory textbox in project explorer, write directory to change to. If the typed directory exists, the program will open it automatically. [OR WILL A PUSH OF ENTER BE ANYWAY REQUIRED TO CHANGE DIRECTORY? SHOULD WE TRY BOTH BEFORE DECIDING?] Pressing esc while in textbox undoes any writing, and reverts back to previous directory.
<b>Scenario b</b>	Click into current directory textbox in project explorer, start writing directory; when autocomplete-results appear below, use up/down+enter or mouseclick to select the directory. Pressing esc while in textbox undoes any writing, and reverts back to previous directory.
<b>Scenario c</b>	Click the down-arrow on the right side of current directory textbox in project explorer, choose desired directory from appearing directory history.
<b>Scenario d</b>	Click "Browse..." in project explorer, use standard [Java] file chooser to select directory to change to.
<b>Precondition</b>	There is no measurement running right now (to prevent the user from getting lost).
<b>Postcondition</b>	The contents of the chosen directory are shown in the project explorer. Last modified project file in the directory will be selected and loaded; it's type (AF/TH) shall be the default type for creating a new project into that directory.
<b>Error condition</b>	If the user has typed an invalid directory and presses enter, a "No such directory" message will be displayed in place of the autocomplete-results.
<b>Goal-derived</b>	5.1.1, 5.1.2, 5.1.3.
<b>Requirements</b>	-

**UC22: Insert/edit project data**

<b>Scenario a</b>	Click any of the project data checkboxes or textboxes (at lower left corner of main window) and enter value.
<b>Precondition</b>	Project file open (usually just created it).
<b>Postcondition</b>	Project data changed, measuring results recalculated, saved automatically to project file. Saving can happen after a short delay to avoid unnecessary file system access.
<b>Error condition</b>	-
<b>Goal-derived</b>	5.1.1 after creating new project.
<b>Requirements</b>	R6, R8, R9

**3.5 Sequences**

*As in automatic demagnetization-measuring sequences (AF sequences), or, thellier temperature sequences.*

**UC23: Insert sequence manually**

<b>Scenario a</b>	Click "Tesla" or "Temp" [SOME BETTER NAMES FOR THESE?] column in the last line (which is empty) of measurement result table, enter value; press down, [tab?] or enter to advance to a new line, enter value; and so on.
<b>Precondition</b>	An open project.
<b>Postcondition</b>	Sequence appended to measurement result table and current project.
<b>Error condition</b>	-
<b>Goal-derived</b>	5.1.3 after creating new project.
<b>Requirements</b>	R6, R12

**UC24: Insert sequence with start-step-stop values**

<b>Scenario a</b>	Click "Start" textfield, insert start value (default value is the last value of the Tesla/Temp column), click "Step" textfield, insert step value, click "Stop" textfield, insert stop value, click "Add sequence" or press enter.
<b>Scenario b</b>	Click "Start" textfield, insert start value (default value is the last value of the Tesla/Temp column), press tab, insert step value, press tab, insert stop value, press enter or click "Add sequence".
<b>Precondition</b>	An open project.
<b>Postcondition</b>	Sequence from <i>start</i> to <i>stop</i> , increasing by <i>step</i> for every step, appended to measurement result table of the current project. Value of "Start" textfield is the previous value of "Stop". "Step" and "Stop" textfields are empty. "Step" textfield has focus.
<b>Error condition</b>	If the last value in the old sequence (excluding completed measurements) is equal to the <i>start</i> value of this new sequence, then start the new sequence from <i>start</i> + <i>step</i> .
<b>Goal-derived Requirements</b>	5.1.3 after creating new project and entering manual values. R6, R12

**UC25: Load sequence**

<b>Scenario a</b>	Click down-arrow in right side of "Load set" combo box; from appearing list, click sequence to load.
<b>Precondition</b>	An open project, at least 1 saved sequence.
<b>Postcondition</b>	Selected sequence appended to measurement result table and selected.
<b>Error condition</b>	(If accidentally inserted the wrong sequence, pressing del will delete it as it's selected after inserting.)
<b>Goal-derived Requirements</b>	5.1.1, after typing project data, load "Basalt" set; 5.1.2 set already loaded for each project file (when created). R6

**UC26: Edit sequence on-the-fly**

<b>Scenario a</b>	Click any unmeasured "Tesla" or "Temp" column in measurement result table, enter new value.
<b>Scenario b</b>	Click any unmeasured line in measurement result table, press del to delete that line.
<b>Scenario c</b>	Right-click any unmeasured line in measurement result table, choose "Delete" to delete that line.
<b>Scenario d</b>	Click-drag any unmeasured line in measurement result table to new position within unmeasured lines.
<b>Scenario e</b>	Select multiple unmeasured lines in measurement result table with shift-clicks and ctrl-clicks, make any of above actions (except enter new value).
<b>Precondition</b>	Unmeasured lines in current project's measurement result table (note that measuring sequence can be going on).
<b>Postcondition</b>	Editing committed, saved to current project.
<b>Error condition</b>	Editing or deleting measured lines won't be allowed.
<b>Goal-derived</b>	5.1.3, when sequence demagnetizes too rapidly, Tomas deletes unmeasured lines.
<b>Requirements</b>	R6

**UC27: Save sequence**

<b>Scenario a</b>	Click "Load set" combo box, enter new sequence name, click "Save set". [2005-03-01 NOT IN UI PROTOTYPE; REMOVE THIS SCENARIO OR DECIDE LATER?]
<b>Scenario b</b>	Right-click any line in measurement result table, choose "Save full sequence...", enter name, press enter.
<b>Scenario c</b>	Select any lines in measurement result table with shift-clicks and ctrl-clicks, right-click on selected lines, choose "Save selected sequence...", enter name, press enter.
<b>Precondition</b>	At least 1 line in measurement results table (although you probably don't want to save a sequence with only one step x).
<b>Postcondition</b>	New sequence set saved to predefined sequence file and available from "Load set" combo box.
<b>Error condition</b>	Ask whether to overwrite, if sequence with the same name already exists (allow to enter new name if choose not to overwrite).
<b>Goal-derived</b>	-
<b>Requirements</b>	-



**UC28: Edit stored sequence**

<b>Scenario a</b>	UC25 "Load sequence" -> UC26 "Edit sequence" -> UC27 "Save sequence" with same name as the loaded sequence.
<b>Scenario b</b>	Click menu "Options"->"Sequences...", edit any sequence in appearing window... [2005-02-25 NOT IN UI PROTOTYPE, MUST BE IMPROVISED :)]
<b>Precondition</b>	At least 1 saved sequence.
<b>Postcondition</b>	Changes saved to predefined sequence file.
<b>Error condition</b>	-
<b>Goal-derived</b>	-
<b>Requirements</b>	-

**UC29: Delete stored sequence**

<b>Scenario a</b>	Click down-arrow in right side of "Load set" combo box; from appearing list, right-click sequence to delete, choose "Delete".
<b>Scenario b</b>	Click menu "Options"->"Sequences...", delete sequence in appearing window... [2005-02-25 NOT IN UI PROTOTYPE, MUST BE IMPROVISED :)]
<b>Precondition</b>	At least 1 saved sequence.
<b>Postcondition</b>	Sequence deleted, changes saved to predefined sequence file.
<b>Error condition</b>	-
<b>Goal-derived</b>	-
<b>Requirements</b>	-

**UC30: Rename stored sequence**

<b>Scenario a</b>	Click down-arrow in right side of "Load set" combo box; from appearing list, right-click sequence to rename, choose "Rename...", enter new name, press enter.
<b>Scenario b</b>	Click menu "Options"->"Sequences...", rename any sequence in appearing window... [2005-02-25 NOT IN UI PROTOTYPE, MUST BE IMPROVISED :)]
<b>Precondition</b>	At least 1 saved sequence.
<b>Postcondition</b>	Sequence renamed, changes saved to predefined sequence file.
<b>Error condition</b>	-
<b>Goal-derived</b>	-
<b>Requirements</b>	-

Identifier	Name	Priority	Set by
R1	SQUID control and usage	1	Customer
R2	Manual control	3	Customer
R3	Calibration reminder	2	Customer
R4	Warning signal	3	Customer
R5	Create project	1	Customer
R6	Autosave project	1	Customer
R7	Load project	1	Customer
R8	Edit project	2	Customer
R9	Recalculate derived measurement data	2	Customer
R10	Append project	2	Customer
R11	Export to other file formats	1	Customer
R12	Create a measurement sequence	2	Customer
R13	Automatic sequence	2	Customer
R14	Single step sequence	2	Customer
R15	Pause automatic sequence	2	Customer
R16	Panic abort	1	Customer
R17	Numeric presentation of measurements	1	Customer
R18	Graphic presentation of measurements	4	Customer
R19	Hotkeys	3	Customer
R20	Change hotkeys	4	Customer
QR1	Error free	1	Customer
QR2	Ease of use	1	Customer
QR3	Quality of documentation	2	Group
QR4	Performance	2	Group
QR5	Expandable architecture	2	Group

Table 1: Requirements summary

## 4 Requirements

Goals of the software set by client, project team and environment. Requirements were collected from 1) powerpoint slides received from client, 2) lectures given by client, 3) user observations held at the squid lab with client (used to collect goal-derived use cases), and 4) user interface prototype demonstrations given to client, after which the client gave suggestions and corrections to prototype functions.

Priorities are from 1 (highest) to 4 (lowest). Our goal is to fulfill at least the requirements with priorities 1 and 2. Requirements with priority 3 will be fulfilled only if time permits, and priority 4's will probably be left out.

Overview of requirements is presented in Table 1.

Requirement format:

**R0: Name of the requirement**

<b>Description</b>	Description of the requirement.
<b>Priority</b>	Requirement priority from 1 (most important) to 4 (least important)
<b>Set by</b>	Who set the requirement; client, project team or environment
<b>Use Cases</b>	Use cases derived from this requirement

## 4.1 Functional requirements

*Define the services which the software offers and how the software behaves, including error handling. May also define things the software shall not do.*

### 4.1.1 Magnetometer

**R1: SQUID control and usage**

<b>Description</b>	Able to control Squid-magnetometer and make measurements with it.
<b>Priority</b>	1
<b>Set by</b>	Customer
<b>Use Cases</b>	UC1, UC2, UC3, UC4, UC5, UC6, UC7, UC8, UC9, UC10, UC11

**R2: Manual control**

<b>Description</b>	Able to operate the magnetometer manually. User can move and rotate sample handler and measure and demagnetize sample.
<b>Priority</b>	3
<b>Set by</b>	Customer
<b>Use Cases</b>	UC11, UC12

**R3: Calibration reminder**

<b>Description</b>	Magnetometer must be calibrated every 24 hours. The program will remind the user when it would be time to do calibration.
<b>Priority</b>	2
<b>Set by</b>	Customer
<b>Use Cases</b>	UC8, UC9, UC10

**R4: Warning signal**

<b>Description</b>	Program should warn user with alarm signal if magnetic field remains on for too long. The implementation depends on if the De-gaussing System's protocol makes it possible.
<b>Priority</b>	3
<b>Set by</b>	Customer
<b>Use Cases</b>	UC2, UC3

#### 4.1.2 Project files

##### R5: Create project

<b>Description</b>	Create a new project file which will include the measurement sequence, measurement results and information about the sample. The file format will be custom made for this program.
<b>Priority</b>	1
<b>Set by</b>	Customer
<b>Use Cases</b>	UC19

##### R6: Autosave project

<b>Description</b>	Program will save measurement data and project information after every measurement step and modification.
<b>Priority</b>	1
<b>Set by</b>	Customer
<b>Use Cases</b>	UC1, UC2, UC3, UC6, UC7, UC8, UC9, UC10, UC11, UC18, UC22, UC23, UC24, UC25, UC26

##### R7: Load project

<b>Description</b>	Saved projects can be loaded into program.
<b>Priority</b>	1
<b>Set by</b>	Customer
<b>Use Cases</b>	UC20

##### R8: Edit project

<b>Description</b>	Ability edit project information and measurement data afterwards. [2005-02-28 WHICH DATA FIELDS?]
<b>Priority</b>	2
<b>Set by</b>	Customer
<b>Use Cases</b>	UC22

##### R9: Recalculate derived measurement data

<b>Description</b>	When the measurement data or the mass/volume of the specimen is changed, recalculate all numbers that have been derived from the modified data.
<b>Priority</b>	2
<b>Set by</b>	Customer
<b>Use Cases</b>	UC22

##### R10: Append project

<b>Description</b>	New measurements can be added to existing projects.
<b>Priority</b>	2
<b>Set by</b>	Customer
<b>Use Cases</b>	UC1, UC2, UC3, UC6, UC7, UC11, UC18, UC20

**R11: Export to other file formats**

<b>Description</b>	Measurements can be saved in .dat, .tdt and .srm files. [2005-03-01 WE DON'T YET HAVE THEIR FORMAT SPECIFICATIONS SO WE CAN'T DO THIS!]
<b>Priority</b>	1
<b>Set by</b>	Customer
<b>Use Cases</b>	UC13, UC14, UC15

**4.1.3 Measurements****R12: Create a measurement sequence**

<b>Description</b>	Able to create AF and Thellier measurement sequences with several different sized steps.
<b>Priority</b>	2
<b>Set by</b>	Customer
<b>Use Cases</b>	UC23, UC24

**R13: Automatic sequence**

<b>Description</b>	Do all the measurement steps automatically.
<b>Priority</b>	2
<b>Set by</b>	Customer
<b>Use Cases</b>	UC3

**R14: Single step sequence**

<b>Description</b>	Do only one measurement step at a time.
<b>Priority</b>	2
<b>Set by</b>	Customer
<b>Use Cases</b>	UC1, UC2, UC6, UC7

**R15: Pause automatic sequence**

<b>Description</b>	Able to stop the measurement sequence after current step. After being stopped, the user can make modifications to the sequence and continue from where the sequence was left.
<b>Priority</b>	2
<b>Set by</b>	Customer
<b>Use Cases</b>	UC4

**R16: Panic abort**

<b>Description</b>	Able to stop any measurement immediately. The demagnetizer will be turned off and the sample holder will stop where it is. The current measurement data will be discarded.
<b>Priority</b>	1
<b>Set by</b>	Customer
<b>Use Cases</b>	UC5

**R17: Numeric presentation of measurements**

<b>Description</b>	Program shows measurement data in numbers. Measurement data will be displayed using scientific notation (1.23e4). [2005-03-01 WHAT ARE ALL THE NUMBERS THAT NEED TO BE CALCULATED? HOW ARE THEY CALCULATED?]
<b>Priority</b>	1
<b>Set by</b>	Customer
<b>Use Cases</b>	UC1, UC2, UC3, UC6, UC7, UC8, UC9, UC10, UC11, UC16

**R18: Graphic presentation of measurements**

<b>Description</b>	Program draws graphs from measurement data. Available graph types are: Stereo Plot, Intensity, Zijderveld, Difference Vectors, Susteptibility, O63, Great Circles. [2005-02-28 WHAT ARE THE DEFINITIONS OF THESE GRAPHS? PRIORITIES PER GRAPH?]
<b>Priority</b>	4
<b>Set by</b>	Customer
<b>Use Cases</b>	UC1, UC2, UC3, UC6, UC7, UC8, UC9, UC10, UC11, UC17

**4.1.4 Others****R19: Hotkeys**

<b>Description</b>	The program has hotkeys for the most often used operations.
<b>Priority</b>	3
<b>Set by</b>	Customer
<b>Use Cases</b>	-

**R20: Change hotkeys**

<b>Description</b>	Possibility to change hotkey assignments.
<b>Priority</b>	4
<b>Set by</b>	Customer
<b>Use Cases</b>	-

**4.2 Quality requirements****QR1: Error free**

<b>Description</b>	The program should not crash. All the functions need to work as documented. The calculations that the program makes from measurements must be done right.
<b>Priority</b>	1
<b>Set by</b>	Customer
<b>Use Cases</b>	-

**QR2: Ease of use**

<b>Description</b>	Program should be easy to use for first time users after it has been explained to them how to take measurements with the magnetometer.
<b>Priority</b>	1
<b>Set by</b>	Customer
<b>Use Cases</b>	-

**QR3: Quality of documentation**

<b>Description</b>	The documentation must be so good that other teams can easily continue the development. Program should have good help pages for the end-users.
<b>Priority</b>	2
<b>Set by</b>	Group
<b>Use Cases</b>	-

**QR4: Performance**

<b>Description</b>	The program must keep up with the SQUID hardware. The user interface should not freeze for periods of over 0,5 seconds.
<b>Priority</b>	2
<b>Set by</b>	Group
<b>Use Cases</b>	-

**QR5: Expandable architecture**

<b>Description</b>	It should be possible to expand the program to process the measurement data in more ways.
<b>Priority</b>	2
<b>Set by</b>	Group
<b>Use Cases</b>	-

**4.3 Environment**

*This section defines requirements for the software which originate from the environment in which the software will be working, and which the software must satisfy for being able to work in its environment.*

Program will be used in normal PC which is connected to magnetometer. The computer that will run this program will be equal or better to 1GHz CPU, 256MB RAM, 1280x1024 resolution. The current computer is running Windows XP.

Taking into account the rapid speed of computer evolution it is possible that the computer in which the program is used can change. Accordingly, the program should be installable by outsiders. We need not prepare to the changing of the magnetometer, as a new magnetometer will probably have its own program. This program does not control the temperature of the magnetometer.

The SQUID hardware and communication with it are described in the section "External interfaces".

## 4.4 Maintainability

It must be possible to continue the development of the program. The documentation must be complete so that other teams can quickly continue the development even if they have not studied user interface designing. It must be possible to add new graph types to the program and export the measurement data to other programs.

## 5 User interface

User interface for the program is loosely generated from goal-derived use cases, which in turn were collected by user observations held with the client at their magnetometer lab (using the old squid software).

Method for generating the user interface is "applied GDD". GDD, or *Goal-Derived Design*, is taught on courses "User Interfaces" and "User Interfaces II" at Department of Computer Science. For more info, see

```
http://www.cs.helsinki.fi/u/salaakso/
Goal-derived design?
Goal-derived use cases?
```

### 5.1 Goal-derived use cases

There were three user observations held, and those generated three goal-derived use cases. The picture sequences of each use case are in the appendices.

In the picture sequences is shown how the program is being used. Every mouse click is indicated with a cursor; follow the white mouse. The typing of text is shown in like manner. Details of the magnetometer measurement sequences have been left out from most of the places to save trees. Only in the first use case, in the first step of the measurement sequence (not calibration), is shown all the phases.

#### 5.1.1 Use case 1: Erkki studies the movement of Antarctica

##### **Erkki's goal:**

To measure where Antarctica was 180 million years ago.

##### **Information:**

- Now is Tuesday 8.2.2005 at 14:27
- Erkki is in laboratory at Physicum

##### **Erkki's knowledge:**

- He has many samples from Antarctica, hand sample, drill's and logbook and data from earlier measurements.
- He has available libra, waterpool and SQUID-equipment and continent draw program
- He can assume good results because sample is Basalt and he knows how basalt usually



is magnetised.

- Hand sample has line which projects to north pole and on drill is same line.

**Logbook:**

Area: Antarctica

Site: Utpostane

Map: 1134

X:

Y:

Latitude: 344

Longitude: -74

Sample: UP-17

Core[ ]/Hand[x]

Rock Type: Basalt

Comments: Dike 10m tos, fine grained, altered?

Orientation:

Date: 14.06.1997

Time: 13:02

Magn. decl:  $289 + -9 = 280$

Dip 14

### 5.1.2 Use case 2: Anomaly in magnetic field at year 500

**Lauri's goal:**

Department of Geophysics' head Lauri Pesonen has been researching samples from Finland around 500 a.d., and measuring them has given some suggestion that there was a peak in Earth magnetic field at that time. Lauri needs more research about the issue, to see if there's some truth in it.

**Fabio's goal:**

Lauri has asked Fabio to study more samples from that time, to be sure about the possible peak in magnetic field 1500 years ago.

**Information:**

- It's Wednesday 9.2.2005 15:05, Fabio is at Geophysics Department's lab in the basement of Physicum, Lauri in his office at third floor
- Fabio has 19 pieces of Estonian jars from year 500, each sample has its own cryptic code name marked on
- Jar pieces are in tubes made of substance X, to make them fit nicely inside magnetometer sample handler
- Estonian samples have already been thellier-measured up to 350 Celsius demagnetization, next up is 380 Celsius.
- Around 10 of the samples are in the oven cooling down from their 50 minute warming up, cooling also takes 50 minutes
- As the other half of samples is cooling, the other is at the same time conveniently warming up

- Results from previous temperatures are in one project file per sample at the Squid-computer
- All samples are supposed to go through the same cycle (which they already have followed up to 350 Celsius): warm up n degrees->cool down without field->measure->warm up n degrees (again)->cool down with field->measure->warm up n+m degrees...
- First one in the next set appears to be sample "pd2-1a"

**Fabio's knowledge:**

- Previous results from Finnish samples suggest that magnetic field was surprisingly high back then
- Fabio can't yet tell what kind of final results to expect

**5.1.3 Use case 3: Tomas studies meteor magnetism**

**Tomas' goal:**

To get some idea how meteor rocks are magnetised.

**Information:**

- Now is Monday 14.2.2005 klo 15:15
- Tomas is in laboratory at Physicum
- Tomas has many samples from meteors and he has splitted some of them into two kind of rocks: old rock pieces and gluerock (space dust?). This particular meteor is from Brazil named Avandahah.

**Tomas' knowledge:**

- Tomas has some insight how meteors are magnetized, but he never can be sure
- Tomas has previously weighted stones: glue 0.0033gm, old 0.0020gm

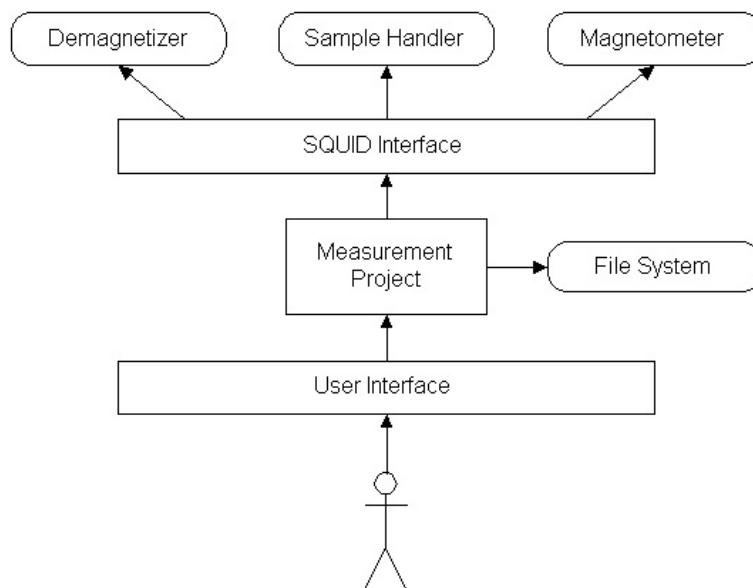


Figure 2: Architecture overview

## 6 Architecture overview

The program architecture can be divided into three parts: SQUID interface, measurement project and user interface. The program will also communicate with the SQUID hardware and local file system. A graphical representation of the architecture can be seen in Figure 2.

SQUID interface is responsible for controlling the hardware in an orderly manner. It will provide high- and low-level controls for using the hardware. Communication with the hardware is done via COM ports. The SQUID interface will hide the protocol-level commands from the programmer and prevent illegal use of the hardware.

Measurement project is responsible for managing the project information, measurement sequence and measurement data. It will receive commands from the user interface and notify the UI when the state of the project changes. It will send commands to the SQUID interface, receive measurement data and save it. When the internal data of the project changes, the copy on the local file system will be automatically synchronized after a short delay (1 second or less).

User interface is responsible for communicating with the user of the program. It will update itself whenever the state of the measurement project is changed. It will send commands from the user to the program.

## 7 External interfaces

*Interfaces to existing software and hardware.*

### 7.1 Existing program

The existing software for using the SQUID is "2G Enterprises Data Acquisition". We have access to the source code for version 2.99.3 of the program. From the old source code we will reuse basically only the SerialIO component. We will build an interface for communicating with the SQUID hardware by using Java and JNI (Java Native Interface).

### 7.2 Hardware control protocols

The SQUID consists of three independent units:

- Automated Sample Handler System (MODEL 2G800)
- Automatic Sample Degaussing System (MODEL 2G600)
- Superconducting Rock Magnetometer (MODEL 755R or 760R)

Automated Sample Handler System controls the movement and rotation of the sample holder. Its protocol is described in the appendices.

Automatic Sample Degaussing System controls the demagnetizer. Its protocol is described in the appendices.

Superconducting Rock Magnetometer reads the measurements from the magnetometer. Its protocol is described in the appendices.

## 8 Validation

In validation it is being checked that a product meets the requirements set for it. Sometimes the requirements have priorities and only the most important requirements are demanded. In document reviews, held between the client and the project team, it will be established that the requirements document and the design document are in line with user requirements and will lead to the production of a program which meets these requirements. Testing shows that the program actually works as specified by the design document. Finally the finished product will be checked against this requirements document.