Testing document x.x

SQUID

Helsinki 15th April 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	31.3.2005	First version (Aki Korpua)
0.2	5.4.2005	Corrections, Use Case sample (Aki Korpua,Mikko Jormalainen)
0.3	12.4.2005	Use case split and now look (Aki Korpua)
0.4	13.4.2005	Integrity test cases (Aki Korpua)
0.5	13.4.2005	Some fixes (Aki Korpua, Mikko Jormalainen)

Contents

1 Introduction

This document describes how this software (Ikayaki) is planned to be tested properly. Mainly this document concentrates on describing methods used for testing and test cases. It is important that all members of team to make tests in same way. This lowers possibility of testing conflicts and helps on integration test phase.

2 Overview of testing the system

Because program will be used to control a magnetometer, testing will be more important than in normal software engineering student projects. We will do unit testing for each class, integrate testing to program and use separate squid emulator to test squid interface system.

In unit testing each class is tested independently. Unit testing will be done by using JUnit. Every programmer will test his own classes. Class should be tested when it is finished and corrected before integration test begins.

Integration testing tests interfaces between classes. It will be done by going through all user interface protos and checking that all sections in requirements document can be done. Some critical sequences which are done many times with program should be done too.

Squid interface integration testing is done simulating real system with emulator. It will be done using Squid-emulator before testing it with real magnetometer. Squid-emulator runs in different machine and is connected by few (2-3) Serial I/O cables. Squid-emulator will be tested with old program (2G) same way before testing Ikayaki-system so that it will have all same tested properties which old program have and both systems have same results with squid emulator.

To verify that old program and new program works same way, we will do critical measurement with old program and emulator, save emulators log file and then use emulator with that log file and do same critical measurement with new program and see that both have same results.

If Rita testing utility is easy enough to use it will be used in testing. Tests will be constructed in such way that every line of code is visited at least once.

3 Testing

3.1 Unit Testing

Unit Testing is done for each class separately. Classes tested with JUnit have it's own ClassNameTest.java class in test-directory and gui-components are tested manually. They should be done before and during coding class. All test cases must be executable after classes are ready. Every class should be tested succesfully before integration tests.

All test cases are listed in sections 4 and 5.

3.2 Integration Testing

Integration testing is mostly done after unit tests are passed for all classes. Testing is done using squid emulator at first and finally with real Squid system. Graphical User Interface is mainly tested in Unit testing and is only looked that all components work together as Squid Interface is tested.

Integration testing for GUI-components is done using top-down method. Testing is done partly during implementation phase, when new components are added to program. After adding component to program it is tested to work properly with other components.

Graphical User Interface is tested using all use cases from 6. If one fails, it is corrected immediatelly and all use cases must be done again. When all use cases are done without errors this phase is ready.

Squid Interface testing is done with Emulator. Use cases are Automatic Measurement, Thellier Measurement and Manual Measurement with all variations (see 6). In first phase we use old program with emulator, save its log file and take results. After that we run it with new program and compare results. If they are not same, corrections are made immediatelly to new program and test is run again. This is done until results are same for all use cases.

3.3 Squid Emulator

Squid emulator is tested with old program. Use cases are Automatic Measurement, Thellier Measurement and Manual Measurement with all variations (see Requirements Document). When all use cases can be done with emulator it is ready enough for integrity testing.

This should be done so that Old Program has same results with SQUID-system and emulator for all use cases. But we don't have resources for this. So this test only tells us that Old and New program works same way with squid emulator.

4 JUnit test cases

Here are listed JUnit test cases for classes.

JUnit is simple java-based framework for testing your java classes. We use it in this project for Unit Testing. For more information visit http://junit.sourceforge.net/.

First you need to download JUnit from http://sourceforge.net/project/showfiles.php?group_id=15278 and extract it to directory (different than java directory) and set classpath for it.

Then you must write test class for every class to be tested. Test classes extend TestCase. They will have test methods, one for each Test Case. Test class also need suite()-method and Main for run. Sample test class:

```
  import java.util.*;
  import junit.framework.*;

  public class SimpleTest extends TestCase {

  //Simple Test Case

  public void testEmptySimple() {
        Simple simple = new Simple();
        //New Simple must be empty
        assertTrue(simple.isEmpty());

  }

  public static Test suite() {
        return new TestSuite(SimpleTest.class);
  }

  public static void main(String args[]) {
        junit.textui.TestRunner.run(suite());
  }
}
```

Type "java junit.swingui.TestRunner SimpleTest" to run test cases for Simple.

4.1 SampleClass

1. Test1

   Conditions:

   - Cond1
   - Cond2

2. Test2

   Conditions:

   - Cond1

5 GUI-component test cases

Here are listed test cases for gui-classes, these are done manually just clicking and changing values.

5.1 SettingsPanel

1. Magnetometer,Handler and Degausser COM-port combobox Conditions:
   - Shows all COM-ports on system in alphabetical order
     Result: success
   - Loads correct COM port from Settings
     Result: success
   - Doesn't allow same value as in Handler and Magnetometer/Degausser COM-port comboboxes
     Result: failed
2. Save button Conditions:
   - Only available if changes are made and values are permissible
     Result: success
   - Unavailable on start
     Result: success
   - On click closes window and saved data correctly
     Result: success
3. Cancel button Conditions:
   - On click closes window and doesnt save changes
     Result: success
   - Always available
     Result: success
4. Calibration constants Conditions:
   - Loads data correctly from Settings
     Result: success
   - Accepts positive and negative decimal numbers
     Tested parameter: -0.430002
     Result: success
5. Degausser ramp Conditions:
   - Loads data correctly from Settings
     Result: success

- Has only elements: 3,5,7,9
  Result: success

6. Degausser delay Conditions:
   - Loads data correctly from Settings
     Result: success
   - Has only elements: 1..9
     Result: success
7. Acceleration and deceleration fields Conditions:
   - Loads data correctly from Settings
     Result: success
   - Accepts only values in range of Integers 0..127
     Tested parameter: 1000
     Result: success
     Tested parameter: -10
     Result: success
8. Velocity and velocity in measurement fields Conditions:
   - Loads data correctly from Settings
     Result: success
   - Accepts only values in range of Integers 50..20000
     Tested parameter: 0
     Result: success
     Tested parameter: 2001
     Result: success
9. Translation positions fields Conditions:
   - Loads data correctly from Settings
     Result: success
   - Accepts only values in range of Integers 0..16777215
     Tested parameter: -1
     Result: success
10. Right limit Conditions:
    - Loads data correctly from Settings
      Result: success
    - Has only elements: plus, minus
      Result: success
11. Rotation field Conditions:

- Loads data correctly from Settings
  Result: success
- Accepts positive and negative Integer numbers
  Tested parameter: -1
  Result: success
  Tested parameter: 0.1
  Result: success

Test results for SettingsPanel26/27

5.2 SampleClass

1. Use case1 Conditions:
   - Cond1
     Tested parameter: param1
     Result: success
   - Cond2
     Result: failed

Test results for SampleClass1/2

6 Intergrity test case

Here is test case which is done in integration phase to all components at same time. This is done in following order and must be repeated until done succesfully. Variations can be made to ensure that all works in different situations.

6.1 Automatic Measurement

1. Open software

   Conditions:
   - Last project is opened, if available.
   - All components are in right place
2. Select Sample holder calibration

   Conditions:
   - opens Sample holder calibration project in Measurement Sequence and Information

- disables Measurement Controls
- enables calibrate button
- disables editing sequence

3. Click Calibration button

   Conditions:

   - Adds new line in Sequence at bottom
   - Starts calibration, progress shown on measurement controls
   - Data is updated correctly in Measurement Details
   - Calibration button changed to stop button

4. Calibration finished

   Conditions:

   - New data added to Measurement Sequence
   - Stop button is changed to calibration button
   - Last modified is updated Sample Holder

5. Select standard sample

   Conditions:

   - opens Standard Sample calibration project in Measurement Sequence

6. Click Calibration button

   Conditions:

   - Adds new line in Sequence at bottom
   - Starts calibration, progress shown on measurement controls
   - Data is updated correctly in Measurement Details
   - Calibration button changed to stop button

7. Calibration finished

   Conditions:

   - New data added to Measurement Sequence
   - stop button is changed to calibration button
   - Last modified is updated in Standard Sample

8. Create new AF project

   Conditions:

   - Calibration project is closed

- new AF project is opened with given name (ProjectInfo,MeasurementSequence)
- Measurement Controls has enabled buttons Measure,Single Step and disable Stop Now!

9. Add project Information

   Conditions:

   - All data is accepted

10. Load Set

    Conditions:

    - Loaded set is correctly added to Sequence

11. Click Measure

    Conditions:

    - Measure button changed to Pause, disables Single Step, enables Stop Now!
    - Rows are highlighted in Explorer and Sequence
    - Starts measuring first phase correctly (no demag now)
    - Picture of demagnetizer is up-to-date
    - Plots are drawn when one phase is over
    - Continues on next phase and does demag first
    - updates Details and Sequence
    - any sequence cannot be changed

12. Pause button

    Conditions:

    - Finish current measure and stops after that
    - Pause button changed to Measure button
    - When finished disables Stop Now!

13. Sequence edit

    Conditions:

    - Steps not yet measured, can be changed

14. Single step button

    Conditions:

    - Performs Next phase on measure
    - Disables Measure button and Single Step button, enables Stop Now!

15. Measure button
    Conditions:
    - Measure button changed to Pause, disables Single Step, enables Stop Now!
    - Rows are highlighted in Explorer and Sequence
    - Continues from Next phase
    - Picture of demagnetizer is up-to-date
    - Plots are drawn when one phase is over
    - Continues on next phase
    - updates Details and Sequence
16. Stop Now! button
    Conditions:
    - Stops measurement immediatelly.
    - Disables buttons and enables Manual Control
    - What now? Is project Manual or is this only temporary?

6.2 Thellier Measurement

1. Open software
   Conditions:
   - Last project is opened, if available.
   - All components are in right place
2. Open existing Thellier project
   Conditions:
   - Project is opened correctly in Information and Sequence
   - Measure button is hidden, Single step enabled and Stop now Disabled
3. Add new line in sequence
   Conditions:
   - It’s inserted correctly.
4. Click Single Step button
   Conditions:
   - Performs Next phase on Sequence
   - Disables Single Step button, enables Stop Now!

- Rows are highlighted in Explorer and Sequence
- Picture of demagnetizer is up-to-date
- Plots are drawn
- updates Details and Sequence

5. Select next Project

   Conditions:

   - Opens project correctly
   - Adds same phase to sequence (as in last project added)

6. Click Single Step button

   Conditions:

   - Performs Next phase on Sequence
   - Disables Step button, enables Stop Now!
   - Rows are highlighted in Explorer and Sequence
   - Picture of demagnetizer is up-to-date
   - Plots are drawn
   - updates Details and Sequence

6.3 Manual Measurement

1. Open software

   Conditions:

   - Last project is opened, if available.
   - All components are in right place

2. Create new AF/Thellier project

   Conditions:

   - new AF/thellier project is opened with given name (ProjectInfo,MeasurementSequence)
   - Measurement Controls has enabled buttons (Measure,) Single Step and disable Stop Now!

3. Adding project data and selecting manual

   Conditions:

   - Data is accepted correctly
   - Manual is enabled

4. Rotate handler

   Conditions:

   - Rotation changes on image of squid
   - Squid-system rotates

5. Move handler to specific place

   Conditions:

   - Image of squid animates correctly and stops on right position
   - Squid-system moves to exact place

6. Demagnetize

   Conditions:

   - Demagnetization is shown on squid image
   - Squid-system demagnetizes

7. Measure-all

   Conditions:

   - Measuring is shown on squid image
   - Data is added to Details correctly (right rotation, background)
   - Data is added to Sequence (only if all degrees done?)

8. Reset measure

   Conditions:

   - Nothing shown, Magnetometer is reseted.