

CS 576 Dependable Software Systems

JUnit Testing Assignment

Please submit a **PDF** file via email. The file should be in the form **id_JUnit1.pdf** (in my case it would be **jhk39_JUnit1.pdf**). You should send the email to jayk@drexel.edu with the subject **CS576 JUnit1 id** (again, in my case the subject would be **CS576 JUnit1 jhk39**). If you have any questions, let me know and I will get back to you as soon as possible.

The assignment will be due in 3 parts:

- Part 1: Monday, July 17th, 2006
 - Part 2: Monday, July 24th, 2006
 - Part 3: Monday, July 31st, 2006
-

Introduction

The purpose of this assignment is to investigate the use of JUnit in the open source world. We will look at open source applications and study their associated test suites. As part of the assignment, you will measure the quality of test suites for an application, develop a test plan, and extend the existing test suite for improved coverage.

Objectives

1. To gain knowledge of developing an automated test suite for a large application.
2. To become familiar with installing and using testing tools.
3. To gain experience writing a test plan.
4. To gain an understanding of how well applications are tested in an open source setting.
5. To gain experience in understanding code written by others.

Part 1: Getting Started

The following describes the tasks necessary to complete this assignment. I recommend starting early in case you have trouble finding appropriate applications.

1. The first step of this assignment requires you to find one application. You can utilize any Java applications that you know of. The requirements of the applications are as follows:
 - a. The source code must be available and written in Java, and the application must be larger than 3000 lines of code.
 - b. Application must have an associated JUnit test suite.
 - c. I recommend finding applications with build.xml files. This will make it much easier to install and execute the test cases.
 - d. Potential places to find open source applications:
 - i. Source Forge: <http://sourceforge.net/>
 - ii. Open Source Software in Java: <http://java-source.net/>
2. Once you find the applications you will need to build the application. After building the application, try out the application to get an idea of what the application is suppose to do.
3. Next run the associated JUnit test suite. The goal is to try to evaluate the test suite. How effective is the test suite? What are the tests testing? Are they good tests?
4. The next task is to try to quantitatively evaluate the test suite and applications through the use of metrics. You will need to find a source code metrics tool. Install the tool, and gather metrics for the applications and the test suites. A few metrics tools are listed below. Feel free to find others.

Report

Finally, create a report that answers the following questions. Organize the report in an efficient manner, such that you describe the steps in installing the tools you needed and describing what they are used for. Be clear, so that anyone can replicate what you did, and understand why you did it.

1. Names of the application
2. Where did the applications come from? url is fine.
3. Why did you choose the application? What does the applications do? A brief description is fine.
4. Briefly describe the process of installing the application. If the application came with an ANT build file. It is appropriate to list the ant commands and their purpose.
5. What metrics tool did you find?
6. What metrics does the tool report?
7. Listing of the output from running the metrics tool on the applications.
8. Listing of the output from running the metrics tool on the test suites.
9. Listing of the JUnit output (text output is fine).
10. In your opinion and from your background in writing software, do software metrics provide any information about code quality? Take a look at the code, what is your opinion of the code quality? What is the correlation with the metrics?
11. Describe the test suite. What kinds of tests were created? i.e., functional tests, most executed portions of the code? common conditions? others?
12. What is the quality of the test suite associated with the applications you found? Characterize the quality in any terms that seem appropriate.

Resources

The following tools and sites may be helpful in your project.

1. JUnit: <http://www.junit.org/>
2. ANT: <http://ant.apache.org/>
3. JDepend: <http://ant.apache.org/>
4. Java Metrics: <http://metrics.sourceforge.net/>
5. CCCC (also analyzes Java): <http://sourceforge.net/projects/cccc>
6. Source Forge: <http://sourceforge.net/>
7. Open Source Software in Java: <http://java-source.net/>