**Joda\_Time Test Plan**

**Team Beck**

**Members:**

Mallepalli Rakesh Reddy

Yashmin Singla

**Collaborator:**

Vignesh Subbian

**Objective:** To develop and execute a software test and/or QA test plan and interpret the test/QA results.

To study a relation between code coverage and effectiveness of test suite.

**Project Overview:** “The standard date and time classes prior to Java SE 8 are poor. By tackling this problem head-on, Joda-Time has become the de facto standard date and time library for Java. The design allows for multiple calendar systems, while still providing a simple API” [2].

**Scope:** To perform tests and analyze the SUT using various tools.

**Out of Scope:** To test its usage and collaboration with other applications.

**Test Environment Details:**

Operating System: Windows 7

IDE: Eclipse

Software: Junit, Pitest, SonarQube, EclEmma, Surefire.

**Test Objectives:**

* To test the project using Junit and to check coverage using EMMA.
* Initially, perform manual code inspection to find any deviation from the standard coding principles and standards.
* Perform Static analysis using SonarQube and compare the results with the manual inspection in order to validate them.
* To use PIT tool to perform mutation testing and validate the effectiveness of the test cases.
* To perform the system testing.
* To study relation between coverage and effectiveness of a test suite.

**Assumptions:**

* We will consider the specifications that we figured out from the code, as a prior list of specifications has not been provided.
* That the application performs and delivers what it has been stated to deliver.

**Effort Estimate:**

* Manual Inspection - 1 week
* Test plan, Initial Testing (unit) and static analysis , Mutation testing- 1 week
* Study relation between coverage and effectiveness of a test suite – 10 days week
* System level testing- 1 week

**Audience:**

To anyone involved in the project, as a member or a collaborator. The execution, to all those who will view it later.

**Exit Criteria:**

Testing the project exhaustively is not an exit criteria as such, but we shall be done with it once we cover the proposed test plan.

**Unit Test Plan:**

**Introduction:** The objective of the unit test is to test the functionality of the individual classes in the SUT and check the coverage using EclEmma. This is the first level in overall sequence of testing. 242 classes from 7 modules are considered for testing.

**Software:** Junit, Surefire.

**Approach:** The 4161 Junit test cases are run on 242 classes. The report is generated using the surefire.

**Pass/Fail criterion:** The test cases are passed it they asserted value and result from executing the function in the class produce the same result, else they are failed.

**Exit criterion:** When all the test cases are successfully run and the report is generated using the surefire and coverage is checked using EclEmma.

**Deliverables:** A HTML report generated using surefire and Coverage report.

**Manual Inspection Plan:**

**Introduction:** The objective of the Manual inspection is to check the code for deviation from coding standards and logic. 58 classes from org.joda.time and org.joda.time.tz modules are considered for manual inspection. This is the second level in overall sequence of testing.

**Software:** Excel.

**Exit criterion:** When all the classes are successfully checked and defects are reported.

**Deliverables:** Two excel sheets reports. (One for each modules), Analysis of results.

**Mutation Testing Test Plan:**

**Introduction:** The objective of the Mutation test is to test the effectiveness of the test suite. Pitest plugin is used to perform mutation test. This is the third level in overall testing process.

**Software:** Pitest, Junit.

**Approach:** 242 classes are mutated using the Pitest and 4161 test cases are run on those mutated classes. Number of mutants killed is calculated.

**Pass/Fail criterion:** A mutant is said to be killed if the test case (at least one) for particular method fails. Mutant is said to survive if none of the test cases for particular method fails.

**Exit criterion:** When test is successfully run and a report is generated.

**Deliverables:** A HTML report, Analysis of results.

**References:** Mutation\_Testing.pdf in Test\_Support folder.

**Quality Assurance Plan:**

**Introduction:** The objective of the QA is to check the quality metrics like cyclomatic complexity and Directory tangle index using SonarQube. This is the fourth level in overall testing process.

**Software:** SonarQube.

**Exit criterion:** When SonarQube is successfully run and the results are analyzed.

**Deliverables:** Analysis of results.

**References:** SonarQube.docx in Test\_Support folder.

**System Test Plan:**

**Introduction:** The objective of the system test is to test the system as the whole. This is the fifth level in overall sequence of testing.

**Approach:**

1. A jar file is generated from the source code.
2. A java project is generated in eclipse and the package containing the example in the source code are added to the project
3. The jar file in the step one is added to the build path of the project.
4. The class files are run individually and outputs are verified manually.

**Exit criterion:** When all the classes are run successfully.

**Deliverables:** The project containing the system tests.

**Unit Test Report:**

**Test Results:**

1. All the 4161 unit tests have passed.
2. The coverage is 89.6%.

**References:**

1. Unit test report: surefire-report.html in Test\_Support folder.
2. Coverage Report: index.html in Test\_Support/EclEmma folder.

**Recommendations:** More Junit test cases can be written to cover the code that has not been covered with the current test suite.

**Manual Inspection Report:**

**Test Results:** 27 major defects and 38 minor defects were found.

**References:** org.joda.time.xlsx, org.joda.time.tz.xslx and Manual\_inspection.docx in Test\_Support folder.

**Conclusions:** These is the summary of defects found.

1. toString method is overwritten in more than 5 classes and hashcode method is not over ridden.

2. Exceptions are neither thrown back nor logged properly in many classes.

3. Many methods have more than permitted 7 parameters.

4. Java coding standards are violated.

**Recommendations:**

1. Follow the java coding standards.
2. Log the errors properly.

**Mutation Testing Test Report:**

**Test Result:** The mutation coverage is 78%

**References:** Mutation\_Report.html , Mutation\_Analysis.docx in Test\_Support folder.

**Conclusions:** All though the line coverage is ~88%, the mutation coverage is just 78% which implies that the quality of the test suite has to be increased. Moreover, 10% of the uncovered mutants suggests that the code coverage has to be increased.

**Recommendations:**

1. The Junit test cases should be written where the mutants are not covered.
2. The Junit test cases should be further refined where the mutants are covered but not killed.

**Quality Assurance Report:**

**Test Result:**

1. Average complexity per method is 2.1
2. There are 3023 defects (static analysis).
3. Direct Tangle Index is 43.3%
4. There are 2740 duplicate lines of code.
5. Comment percentage is around 50.

**References:** SonarQube.jpeg in Test\_Support folder.

**Observations and Recommendations:**

1. Cyclomatic complexity of many functions is above 10. Therefore this has to be reduced for easy maintenance and testing of the code.
2. Many Methods have more than 7 parameters. These has to be reduced.
3. The Directory tangle index has to be reduced to enable easy debugging and change cycles of the code.

**System Test Report:**

**Test Result:** All the tests have passed and results are obtained as expected.

**Recommendations:**  The system tests written are not in the format of unit tests and have to be manually verified, which is difficult as the number of system tests increases. Therefore it is better to automate them.

**Joda-Time Test Report:**

**Overview:** All levels of testing are performed according to the plan (Except the study). The results of the tests are satisfactory.

**Test Results:** *Refer to the individual level test reports for detailed description.\**

1. The unit testing gave 100% pass rate.
2. The line coverage was 88%.
3. The mutation coverage was 78%.
4. System testing was fully successful.
5. The QA showed that the code is ~50% comments and defect density is 1.12/class.

**Joda-Time normal and i18n version differences:**

The basic version of the Joda-Time is developed tested and built into a jar file (Which has all generic features like time, offset, Formats etc.). Now an i18n version is developed over this basic version providing extra details for each zone.

**Conclusions and Recommendations:** Since the SUT has passed all the levels of testing satisfactorily, it can be released with minor modifications.