

**Sri Lanka Institute of Information Technology**

**Group Assignment**

Software Engineering Process & Quality Management (SE 3010)

Year 03 Semester 01 – 2017

Group ID : 03 (WE Batch)

Submitted by:

1. IT 14098888 - S.C.G. Liyanage
2. IT 14084614 - D.A. Polwattage
3. IT 14104640 - S.R. Rajapakse
4. IT 15068774 - A.P.I.R. Jayathilaka

Submitted to:

…………………………..

Mr. Indraka Udayakumara / Mr. Saman Gunawardena

20-05-2017

**Table of Contents**

[Introduction 2](#_Toc483062569)

[1.0 Static Code Analysis – Kiuwan 3](#_Toc483062570)

[1.1 Introduction to tool 3](#_Toc483062571)

[1.2 Download and configuration details 3](#_Toc483062572)

[1.3 Usage of Tool 3](#_Toc483062573)

[1.4 Strengths 3](#_Toc483062574)

[1.5 Weaknesses 3](#_Toc483062575)

[1.6 User Reviews of the tool 3](#_Toc483062576)

[1.7 Snapshots of tool usage using selected code samples 4](#_Toc483062577)

[1.8 Outcome of the tool 4](#_Toc483062578)

[2.0 Structural Code Coverage – Jacoco 5](#_Toc483062579)

[2.1 Introduction to tool 5](#_Toc483062580)

[2.2 Download and configuration details 5](#_Toc483062581)

[2.3 Usage of Tool 5](#_Toc483062582)

[2.4 Strengths 5](#_Toc483062583)

[2.5 Weaknesses 5](#_Toc483062584)

[2.6 User Reviews of the tool 5](#_Toc483062585)

[2.7 Snapshots of tool usage using selected code samples 5](#_Toc483062586)

[2.8 Outcome of the tool 6](#_Toc483062587)

[3.0 Bug Tracking Tool – Jira 7](#_Toc483062588)

[3.1 Introduction to tool 7](#_Toc483062589)

[3.2 Download and configuration details 7](#_Toc483062590)

[3.3 Usage of Tool 7](#_Toc483062591)

[3.4 Strengths 8](#_Toc483062592)

[3.5 Weaknesses 8](#_Toc483062593)

[3.6 User Reviews of the tool 9](#_Toc483062594)

[3.7 Snapshots of tool usage using selected code samples 9](#_Toc483062595)

[3.8 Outcome of the tool 10](#_Toc483062596)

[Conclusion 11](#_Toc483062597)

[References 12](#_Toc483062598)

# Introduction

Testing tools are used to test the code in order to increase the code quality and produce a quality software to the end user. There are tools to test the code in different ways such as static code analysis, structural code analysis and bug tracking. Static code analysis is a method of computer program debugging that is done by examining the code without executing the program. For an example, Kiuwan is a tool which does the static code analysis. Structural code analysis is the determination of the effects of loads on physical structures and their components. Jacoco is an example for structural code analysis. And bug tracking or defects tracking system is a software application that keeps track of reported software bugs in software development projects. JIRA is the most popular example for bug tracking tool.

# Static Code Analysis – Kiuwan

## 1.1 Introduction to tool

Kiuwan is a static code analysis tool for code quality management which is offered as software as a service (SaaS) product. [1]

## 1.2 Download and configuration details

Basically, two download options [2]

1. Free trial: - If it is for non-business purpose, have to go with the Github free repo which can be used as a free open source software. For scanning our code base, it should be in the same Github account.
2. Local analyser

## 1.3 Usage of Tool

* CIOs need this to make strategic decisions to improve software development.
* ForQA managersand engineers who need actual state of applications under development.
* CSOs to tackle security from the application perspective.
* Project managers can know the health of the projects from a technical perspective.
* Application architects can discover structural flaws early in the development process.
* Developers want this to develop the best software possible and learn as much as possible in the process.
* Refer [3]

## 1.4 Strengths

* It has the right information to the right stakeholder in the IT department. [3]
* Supports for all major programming languages like Java, C/C++, JavaScript, PHP, C#, .NET, VB, Objective-C, Cobol, SQL, etc. [1]

## 1.5 Weaknesses

* When using the Github as an open source Kiuwan, it can analyse only the repositories which are in the Github account.

## 1.6 User Reviews of the tool

*“Integration with Jenkins and JIRA, And The Security Support, Are Valuable.” -* [Jorge Para Molina](https://www.itcentralstation.com/users/jorge-para-molina) [4]

## 1.7 Snapshots of tool usage using selected code samples

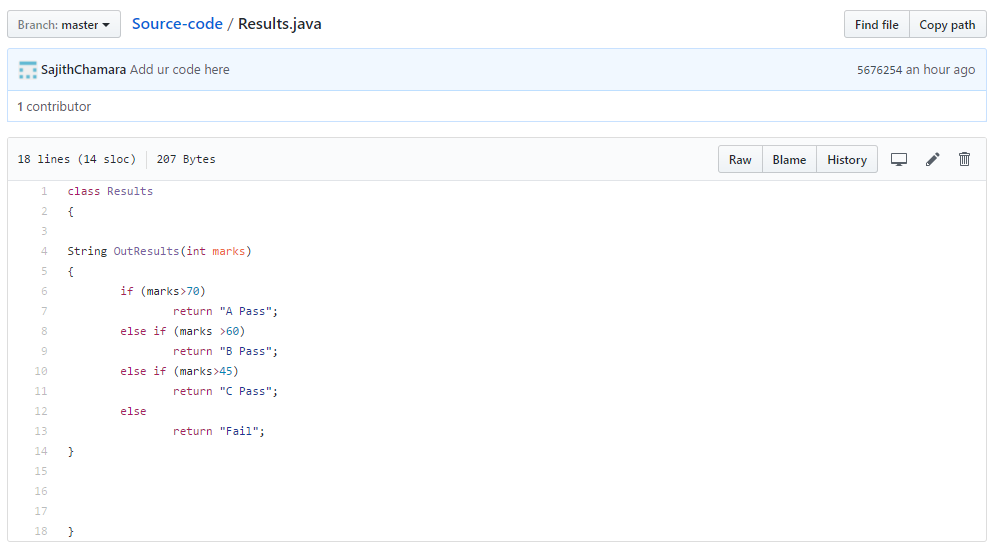


Figure 1.1:- Results.java in Github

## 1.8 Outcome of the tool

Figure 1.2:- Kiuwan report

# 2.0 Structural Code Coverage – Jacoco

## 2.1 Introduction to tool

Jacoco is a structural code coverage tool to generate a report for how much percentage of java code is covered by implemented test methods.

## 2.2 Download and configuration details

Download from [5] or mention the URL in the root pom.xml to download through internet. Build the java project with test classes, using maven or ant. Then render the index.html page which is in the <YourProjectName>\target\site\jacoco to see the generated report.

## 2.3 Usage of Tool

Developers are massively used in maven projects to see the code coverage level.

## 2.4 Strengths

Easy to use and view a clear report.

## 2.5 Weaknesses

Up to now only supported to java projects which built using maven or ant.

## 2.6 User Reviews of the tool

Refer [6]

## 2.7 Snapshots of tool usage using selected code samples

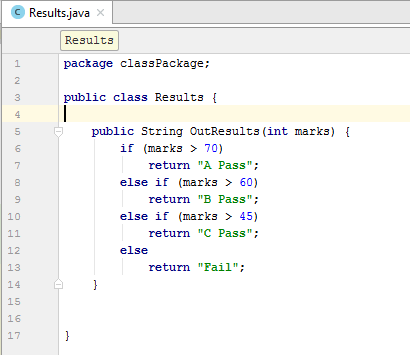
****

Figure 2.1:- Results.java

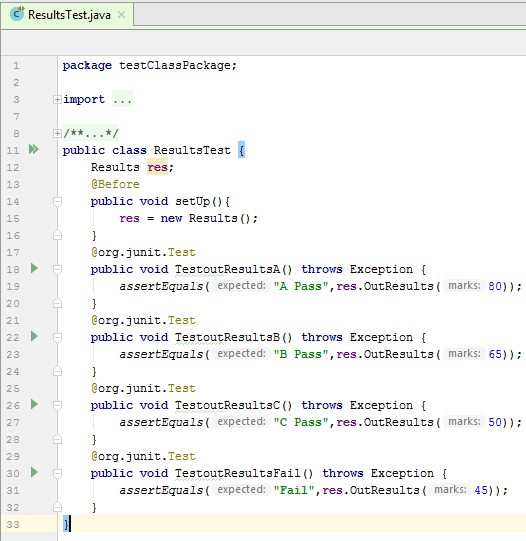
****

Figure 2.2:- ResultsTest.java

## 2.8 Outcome of the tool

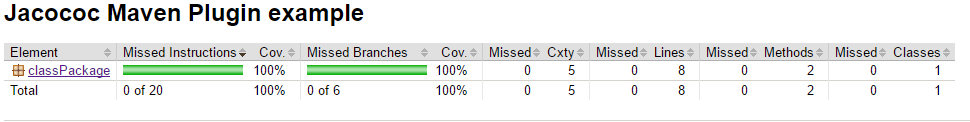
****

Figure 2.3:- JaCoCo Report

# 3.0 Bug Tracking Tool – Jira

## 3.1 Introduction to tool

Jira is a defect tracking/project management tool by Atlassian, Inc., the current version is 6 which is platform independent.

## 3.2 Download and configuration details

* Creating a trial account

1. Go to <https://www.atlassian.com/software/jira/try>
2. Select a package

Enter the details as follows,

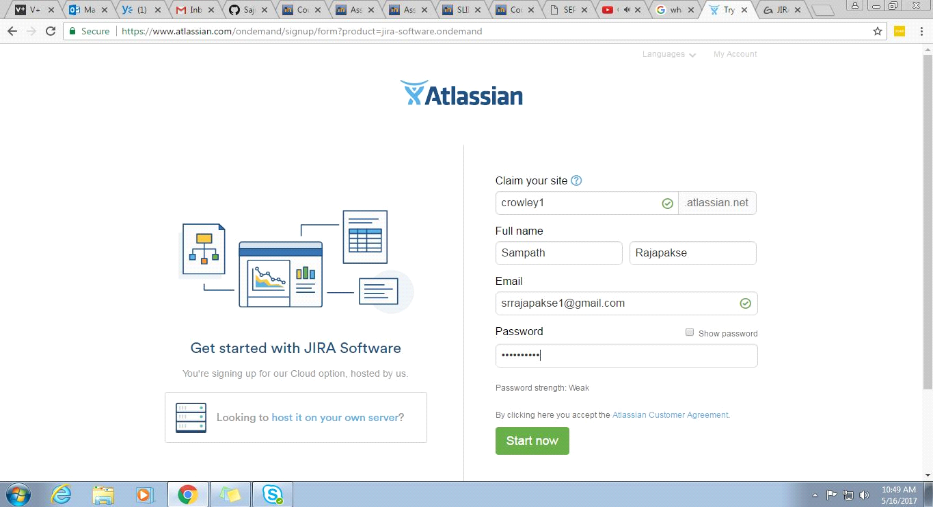


Figure 3.1 – Creation of Jira Account

## 3.3 Usage of Tool

Software project development teams, help desk systems, leave request systems, QA teams widely used for bug tracking, tracking project level issues- like documentation completion and for tracking environmental issues. A working knowledge of this tool is highly desirable across the industry.

* Basics about JIRA:

JIRA in its entirety is based on 3 concepts.

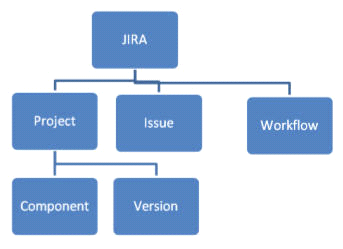


Figure 3.2 – Jira Basics

1. Issue: Every task, bug, enhancement request; basically anything to be created and tracked via JIRA is considered an Issue.

2. Project: a collection of issues

3. Workflow: A workflow is simply the series of steps an issue goes through starting from creation to completion.

Say the issue first gets created, goes to being worked on and when complete gets closed. The work flow in this case is:

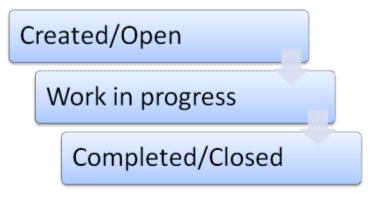


Figure 3.3 – Jira WorkFlow

## 3.4 Strengths

* Flexible - JIRA started out as being a bug tracking system. It has become a general purpose issue/task tracking solution. You can use it to track any "tasks".
* Customizable - Every aspect of the system can be customized through configuration (not custom development). It does add a bit complexity to the JIRA admin, but it is still a better trade off than writing code.
* Workflow – JIRA has a robust and customizable workflow. Workflow can be designed graphically.
* Linking: You can link different tasks. For example, TASK-123 is depended on TASK-256. you can be aware of the dependency among the tasks.
* Sub-Task: A complex task can be divided to smaller sub-tasks so that it is more manageable and can be assigned to different people.
* Refer [7]

## 3.5 Weaknesses

* Hard to setup- JIRA is a large, complicated piece of software that started as a bug tracker and evolved into much, much more.
* Difficult to use- JIRA’s UI is not exactly user friendly, which is, once again, an outcome of being a feature-heavy one-size-fits-all solution. For many usage scenarios you’ll require plugins, which add to the overall UI clutter. JIRA requires time to learn and use effectively
* Refer[8]

## 3.6 User Reviews of the tool

* *“Jira has limited support for other phases of agile like retros, retro actions, demo feedback etc. which are also essential part of agile development”*-Gaurav Raman-Scrum Master/System Developer at H&M(Consultant).
* *“JIRA isn't one of the easiest tools, but it is more comprehensive than others - that's why it's that popular. Especially when it's integrated with time tracking software - TimeCamp seems to be a perfect solution*.”- Aleksandra Rybacka-Copywriter, Content Designer, Social Media Specialist / Hackyourlife.pl.

## 3.7 Snapshots of tool usage using selected code samples

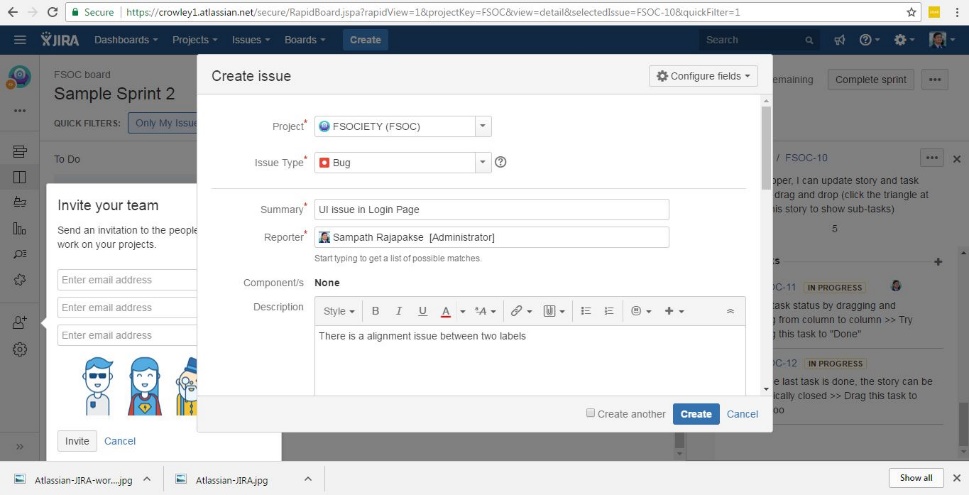
****

Figure 3.4 – Creation of an issue.

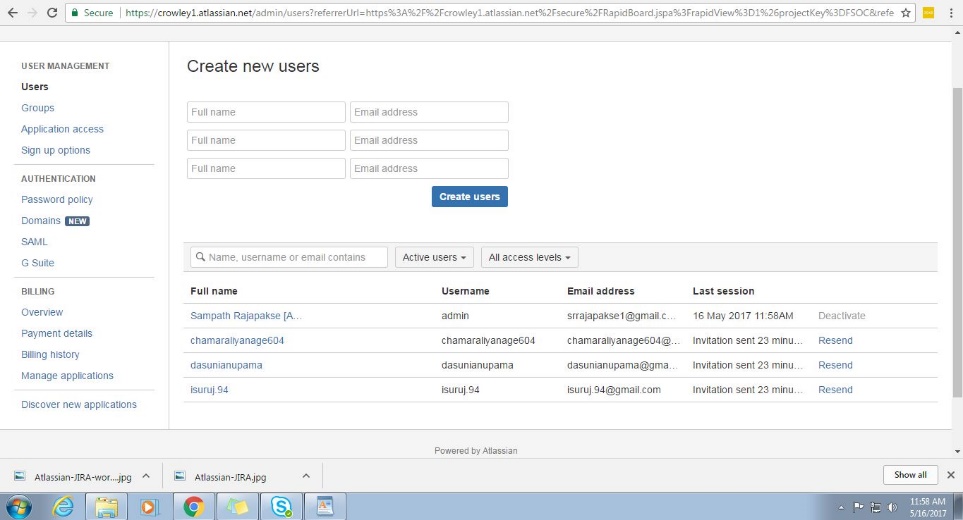
****

Figure 3.5 – Adding users to projects.

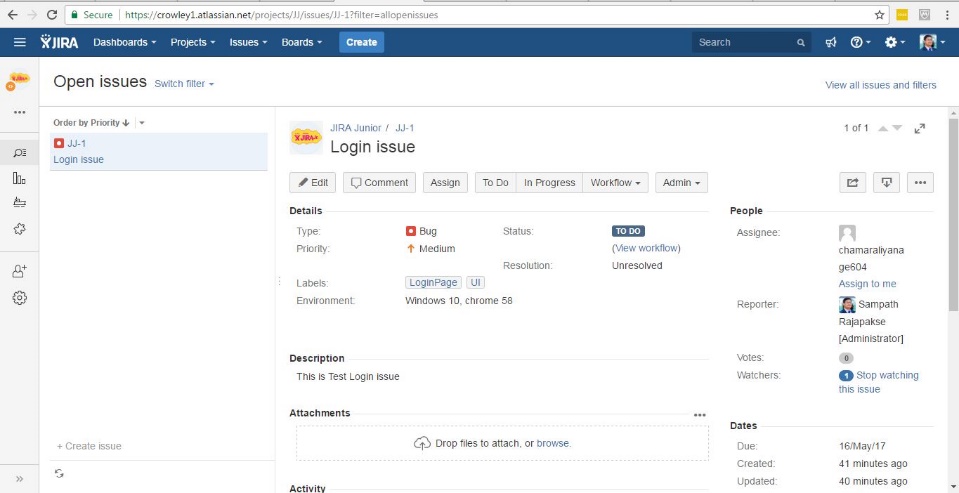
****

Figure 3.6 – Issue after creation.

## 3.8 Outcome of the tool

* Can track issues(bugs) in the development project.
* Can generate various software development related reports and charts (Burn down, Sprint reports).
* Helps in the development phase from start to finish.

# Conclusion

As we have examined the static code analysis, structural code analysis and bug tracking system tools, those tools seemed to have unique strength and weaknesses for each. Some tools cover lines of the code which tested, number of defects for a system, test the code quality by checking for syntax etc. By using these tools developers and quality assurance engineers can easily identify the errors and correct them and increase the product quality as well as code quality. Reduce time consuming, increase the performance and efficiency, improve the code quality are some of advantages of using these testing tools. Specific to particular language, have to use more than one tool to test the software are few disadvantages of using testing tools. Overall it will help to increase the product quality at the end.

# References

[1]"Technical Debt Computation: Comparison Of Top Tools". *Build Blog by ThinkApps | Content on Entrepreneurship, Mobile Apps, Web Platforms and more*. N.p., 2017. Web. 20 May 2017.

[2]"Software Analytics Platform - Kiuwan". *Kiuwan*. N.p., 2017. Web. 20 May 2017.

[3]"Who Is It For? - Kiuwan - Kiuwan Documentation". *Kiuwan.com*. N.p., 2017. Web. 20 May 2017.

[4] "Product Review For Kiuwan". Itcentralstation.com. N.p., 2017. Web. 20 May 2017.

[5] "Eclemma - Jacoco Java Code Coverage Library". Eclemma.org. N.p., 2017. Web. 20 May 2017.

[6] Lakshmanan, View. "Which Code Coverage Tool To Choose?". tier1app. N.p., 2017. Web. 20 May 2017.

[7] Guo, Charles. "The Advantages Of JIRA". *linkedin*. N.p., 2016. Web. 20 May 2017.

[8] Pracuch, Piotr. "5 Reasons NOT To Choose Atlassian JIRA For Agile Projects". *linkedin*. N.p., 2015. Web. 20 May 2017.