

JIRA in Agile Scrum to Improve Cycle-Time Reduction – A Special Reference to Software Testing

[Rosnisa Abdull Razak]

Abstract — Scrum methodology is an Agile management method approach toward software development as it progresses incrementally and repetitively. JIRA helps project team member tracking and managing the requirements, test cases and bugs that emerge during a project is a critically important task, but one that few teams do effectively. In this paper we will discuss, implementation of JIRA as part of Test Management tool in Agile Scrum project to improve cycle-time reduction.

Keywords — JIRA, Agile Scrum Development Life Cycle, Test Management Tool, Improve cycle Time Reduction.

I. Introduction

JIRA is a propriety issue tracking product, developed by Atlassian, used for bug tracking, issue tracking, and project management [1]. JIRA helps project managers stay on top of what's happening, define and manage processes, and ensure task ownership and completion. JIRA is a flexible web-based issue tracking, task tracking, and project management software solution. JIRA can be customized with different fields and workflows to manage multiple projects, users and groups. The reporting features in JIRA apply equally well to how many issues have been closed or how many ideas or strategies have been implemented.

Why we use JIRA for issues tracking? It is because JIRA provides a way to track issues collaboratively, allowing users within a project to easily access information on all issues at a central location. JIRA's focus on task achievement by managing all projects and their many components in one spot, keep team member on track, keep team member stay informed of project progress and access securely online.

Agile development methodology provides opportunities to assess the direction of a project throughout the development lifecycle. This is archived through regular cadences of work known as sprints or iterations, at the end of which teams must present a potentially shippable product increment [2]. Scrum is the most popular way of introducing Agile due to its simplicity and flexibility. Scrum emphasizes empirical feedback, team self-management, and striving to build properly tested product increments within short iterations.

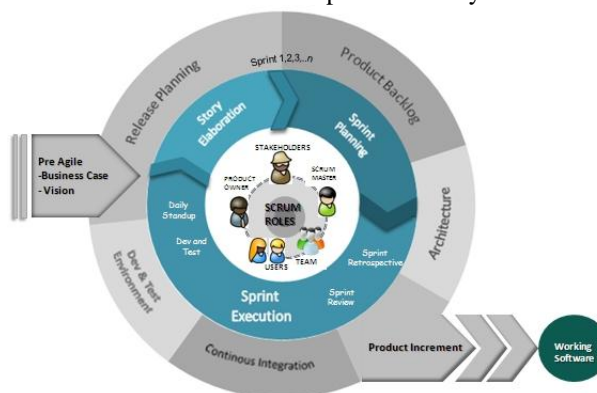
The organization of this paper is as follows. In section II, will describes the background of Scrum Development Life Cycle. In section III, we will discuss on JIRA tracking tool that will improve cycle-time reduction. Next, section IV we discuss the evaluation of using JIRA in agile project. Finally, we conclude our study in section V.

II. Agile Scrum Development Life Cycle

In this section, we will discuss on Agile Scrum Development Life cycle in details from pre-agile, release planning, story elaboration, sprint planning, sprint execution, sprint review and sprint retrospective.

What is Scrum? Scrum is an iterative, incremental framework for project management and agile software development. It is a process skeleton which contains sets of practices and predefined roles. [4] Scrum enables the creation of self-organizing teams by encouraging co-location of all team members, and verbal communication across all team members and disciplines that are involved in the project. Refer figure 1 for Scrum Development Life Cycle [3].

FIGURE 1
Scrum Development Life Cycle



A. Pre-Agile

The purpose of pre-agile is to establish business case which envisions the product from business perspective and alignment with key stakeholders. It is important to ensure project readiness and commitment to adopt agile methodology.

Rosnisa Abdull Razak (Author)
Product Quality and Reliability Engineering
Mimos Berhad
Malaysia

B. Release Planning

The purpose of release planning is to establish a release strategy that the Scrum Teams and the rest of the organizations can understand and communicate which layout the overall project. The release strategy is then used to create sprint plans for each individual sprint. Release planning meeting will be conducted to prioritize product backlog based on complexity and business value.

C. Story Elaboration

The purpose of story elaboration is to establish sprint goal, select and elaborate user stories for upcoming sprint. The team will define a sprint goal, which is a short description of what the sprint will attempt to achieve. The team then will select user stories from product backlog to be implemented in the upcoming sprint. The product owner described the selected user stories in detail including the basic flow and acceptance criteria.

D. Sprint Planning

The Sprint Planning is time boxed to 2-4 hours, and shall be attended by scrum team member. During the meeting the team figures out how it will turn the user stories into a done increment. Each user story is broken down into smaller manageable tasks, no longer than 16 hours in duration, and assigned to developers.

E. Sprint Execution

The purpose of sprint execution is to implement the selected user stories by executing sprint backlog and validate it against sprint goal. Functionality is developed and consolidated into build as per schedule, which is deployed into the test environment. As soon as a user story is deemed developed, it is immediately tested. If changes or improvements are identified during testing, they can be included in the next daily sprint cycle.

F. Sprint Review

The sprint review meeting is time boxed to 2 hours. The purpose of the sprint review is for the team to present to the product owner and the stakeholders functionality that is done. Meaning of 'done' is means that the functionality is completely engineered and could be potentially shipped or implemented.

G. Sprint Retrospective

After the sprint review and prior to the next sprint planning meeting, the scrum team has a sprint retrospective meeting. This is final activity in sprint and purpose is to inspect how the last sprint went in regards to people, relationships, process and tools. The inspection should identify and prioritize the major items that went good and bad during the sprint.

III. Implementation – JIRA as a Test Management Tool

Test management most commonly refers to the activity of managing the testing process. A test management tool is software used to manage tests (automated or manual) that have been previously specified by a test procedure. Test management tools often include a requirement specification management module that allows automatic generation of the requirement or traceability.

Test Management Tools are used to store the information how the testing is to be done, plan the testing activities and report the status of quality assurance activities. A test management tool that includes everything needed to manage the test process can save the hassle of installing separate applications that are necessary for the testing process. A test management tool tracks bug status, defects and projects tasks, and allows for collaboration across the team.

Test management encompasses anything and everything that we do as testers. Our day-to-day activities include:

- Creating and maintaining release/project cycle/component information
- Creating and maintaining the test artifacts specific to each release/cycle that we have-requirements, test cases and etc.
- Establishing traceability and coverage between the test assets
- Test execution support – test suite creation, test execution status capture
- Bug tracking/defect management

In this section, we will explain and discuss more how we used JIRA as a Test Management Tool to in order to improve cycle-time reduction.

A. Creating a Test Case

User Story is created in Release Planning stage and will be elaborate during Story Elaboration stage. Test Case is creating during Sprint Planning. Before JIRA is introducing, developer is using Microsoft Word document to create and update the requirement. After the requirement (User Story) is finalize then Software Configuration Manager (SCM) will do the tagging in the requirement.

Tagging is the unique identifier for each requirement in a project. We want to see the relationship and traceability between requirement and test cases and to make sure all the requirements are tested. One of the problem of using Rational RequisitePro is tester is too dependable to SCM.

Before JIRA is popular in Agile Scrum projects, we are creating test cases by using Microsoft Word. It then become a problem due to different tester may have different type of test

case templates and procedures. Then after the test cases are reviewed and approved, then tester will upload the Test Case in Rational RequisitePro for tagging.

The reason is, SCM will do the tagging and it takes 1 or 2 days depending on the SCM workload. After JIRA is introducing, things become easy because no more dependable to SCM for tagging. Tester can link test case to requirement directly.

Run the Test Case	No activity is trace when we run test case manually.	Activity will be trace in JIRA and tester will update effort estimate to run the test cases.
Update Test Result (pass/fail)	Test result will be updated in separate excel template.	Tester will update test result pass or fail.

TABLE 1

Comparison on JIRA Test Case Management with MS Word and RequisitePro Test Case Management

Area	MS Word + RequisitePro Test Case Management	JIRA Test Case Management
Create Test Case	1. Create test case in Microsoft (MS) Word template. 2. Manually key-in the Test Case number.	1. Create test case in JIRA test case template. 2. Test case number is created upon saving the test case.
Review Test Case	Tester compiles all the test cases and email to team member for review. We will have to set a review meeting with team member.	No need to email. With JIRA test case management, we can assign the test case for review. Tester and developer will then receive email notification then will review and approve the test cases.
Traceability from requirement to Test Case	Tag the traceability from requirement to test case in Rational RequisitePro is requires more time because we need to upload the test cases then we have to do tagging one by one.	When we create the test case, we can link the test case to requirement. Refer to figure 2 to see the traceability from Test Case to User Story.
Baseline	After the test cases is reviewed and tagged then tester will baseline the test cases in SVN.	No need baseline everything will be in JIRA.

FIGURE 2

Example of User Story in JIRA

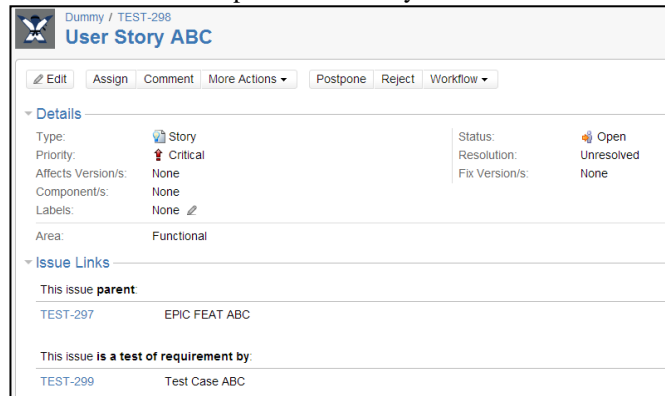
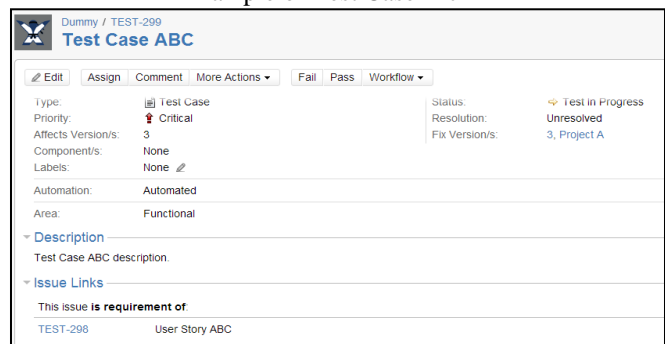


FIGURE 3

Example of Test Case in JIRA



B. Creating a New Issue

What an issue is? An issue is anything that you would track to completion example a document to be created, a document to be reviewed, a bug or an environmental issue. Before JIRA, we are using excel to keep track of all issues and Rational ClearQuest to raise issue or bug.

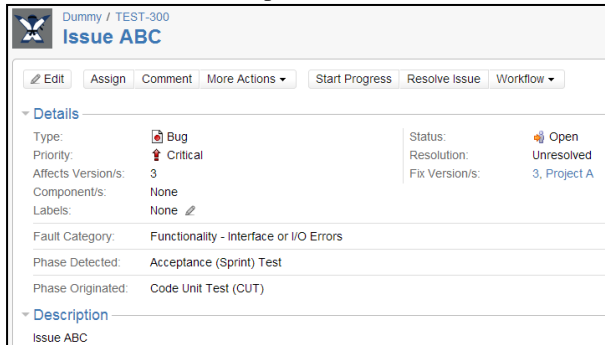
TABLE 2

Comparison JIRA Issue Tracking with Excel and ClearQuest Issue Tracking

Area	Excel + ClearQuest Issue Tracking	JIRA Issue Tracking
Create Issue	Create issue in ClearQuest.	Create issue in JIRA.

Issue Tracking	Assign issue to developers.	Assign issue to developers.
Issue Reporting	Issue reporting will be done in Excel where tester has to calculate manually how many issues categorized by severity.	Issue reporting is done in JIRA. Tester can filter the issue to calculate issue categorized by severity.

FIGURE 4
Example of Issue in JIRA



	<p>Test cases creation per day using JIRA = 20 test cases</p> <p>Test Case creation using WORD doc 1 TC = 48 min 85 * 48 = 4080 min = 68 hours (8.5 days)</p> <p>Test Case creation using JIRA 1 TC = 24 min 85 * 24 = 2040 min = 34 hours (4.3 days)</p> <p>Reduction time: = 68 - 34 = 34 hours (4.3 days)</p>
Issue Tracking	<p>Total Problem Request (PR) and Change Request (CR) raised: 39 in JIRA</p> <p>Average 3 min faster per PR+CR raised compared against CQ. The average obtained through a simulation between JIRA and CQ comparison.</p> <p>Reduction time: = 39 * 3 = 117 min = 1.95 hours</p>

IV. Evaluation – JIRA Improve Cycle-Time Reduction

One of the challenges with agile methods is to measure process improvements. How do we use cycle time as a meaningful measurement? Cycle time is the total time from the beginning to the end of your process, as defined by you and customer. Cycle time includes process time, during which a unit is acted to bring it closer to an output, and delay time, during which a unit of work is spent waiting to take the next action. [4]

Cycle time is the time spent working on an issue; the time taken from when work begins on an issue to when work is completed, but also includes any other time spent on the issue. For example, if an issue is reopened, worker on, and completed again, then the time for this extra work is added to the cycle time.

Refer to Table 3 for cycle time reduction calculation. From the table below, we know that by using JIRA it will improve cycle time reduction on Test Case creation and issue tracking.

TABLE 3
Cycle-Time Reduction

JIRA	Cycle-Time Reduction
Test Case Management	<p>Total Test Cases: 85 in JIRA</p> <p>Test cases creation per day using WORD document = 10 test cases</p>

V. Conclusion

Cycle time reduction has many positive effects on the creating issue, creating test case and test reporting process from reducing work in process to increasing throughput and all without increasing the resources required.

We have found JIRA to be the single great tool for nearly all parts of traditional bug tracking tool. JIRA is capable issue management application especially designed for system and software application developers. Over the years, JIRA has grown into a capable workflow management application that can be used to manage just about any business process that can benefit from workflow automation.

JIRA really helps to improve cycle-time reduction in test and reporting. The conclusion, JIRA is certainly usable as a test execution management environment. The more the system is automated, the less time staff spends on the document writing and completion of project.

Acknowledgment

The authors wish to thank the other members of the Product Quality and Reliability Engineering (PQRE) groups at Mimos Berhad for their help throughout the data collection for Cycle-Time reduction for Test Case and Issue Tracking.

References

- [1] <http://en.wikipedia.org/wiki/JIRA>
- [2] <http://agilemethodology.org/>
- [3] MIMOS Scrum Development Life Cycle - mimos-spp-poc-agile_framework
- [4] http://www.isixsigma.com/dictionary/Cycle_Time-217.htm

About Author (s):



[“If you don’t care about quality, you can meet any other requirement” - *Gerald M. Weinberg*]