2011 5th Malaysian Conference in Software Engineering (MySEC)

Using JIRA and Redmine in Requirement Development for Agile Methodology

Hasliza Md Sarkan, Tengku Puteri Suhilah Ahmad, Azuraini Abu Bakar
Software Development
MIMOS Berhad
Kuala Lumpur, Malaysia

Abstract—Today, Agile development has become popular due to the ability to deliver fast and hence reduce cost. It is expected that by 2012, Agile development method will be utilized in 80% of all software development projects. Since 2010, some of software projects in MIMOS have adopted agile methodology. Therefore, this paper will focus on how agile methodology applied in MIMOS has influenced the requirement process. The paper will discuss on the benefits and challenges that MIMOS faced using JIRA and Redmine in requirement development.

Keywords - JIRA; Redmine; Requirement Development; Agile

I. INTRODUCTION

Agile development methodology is based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing and cross-functional teams. It breaks the tasks into small increments where the workable solution is achieved at the end of the iterations, so-called sprints. In each iteration, a team works through a full software development life cycle (SDLC) including planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the sprint, the workable solution is demonstrated to product owners.

Adoption of Agile development in MIMOS has helped the team to adapt pro-actively to the changes and understands the market needs better. By developing the solution by parts, it allows the team to have adaptability to requirement changes and ability to improve the solution.

This paper is organized as follows. Next section discusses on how Agile is being implemented in MIMOS. Section III presents the advantages and disadvantages of using JIRA and Redmine. Section IV explains the benefit to MIMOS in implementing Agile. Finally, in Section V is the conclusion of the study.

II. AGILE IMPLEMENTATION IN MIMOS

A. Agile Establishment

Today's software development is impacted by the rapid technology change. Requirements are expected to be changed to align with the high demand in the markets [1]. Companies are also expected to develop the system in a shorter timeframe as to keep up with the customers' expectation.

With the current market evolution, implementation of Agile in MIMOS was initiated to support the organization in becoming capable and competent in adopting and serving the rapid market needs. Out of several Agile methodology such as Extreme Programming, Scrum, Lean Management, Scrum was selected. Scrum was preferred because of the flexibility in managing the development lifecycle. It does not dictate much on the engineering processes. It simplifies the traditional development processes by speeding up the development cycle by separating the deliverables in a few iteration, so-called sprints.

In MIMOS, Agile Development was implemented as one of the process improvement initiatives so called Rapid Development Process. MIMOS Agile process framework was introduced to support the implementation by detailing out the process flow, templates and guides on how Agile to be conducted. The motivation behind this is to allow practitioners and projects to build on the practices as team becomes more familiar and mature with Agile principles.

Product backlog, sprint backlog, release planning, user stories, and sprints are few concepts in Scrum which were slowly introduced. Teams have gradually adopted and understand the concept of 'user stories' where it describes the requirement definition in terms of customers' perspective in short, plain-language description. This approach helps to think about whom a certain feature is built for and why, and as a result is the approach that typically prefer to take. The user stories topic will be described more in the next topic; Agile Requirement Development.

B. Project Characteristics

Initially, each project is evaluated whether it is suitable to practice Agile methodology. Based on the project criteria, the project team is assessed prior to adopting Agile methodology. Below is the key characteristic for the project to adopt Agile implementation:

1. Product owner have been identified and willingness to participate in the project is important. They are responsible to validate the requirement scope.
2. The high-level requirement has been identified. At least the scope has been agreed during the contract book signing. From the product backlog, the overall effort required to execute the project can be estimated.

978-1-4577-1531-0/11/$26.00 ©2011 IEEE

3. Scrum master must have technical background as well as good influence to the team members.
4. Scrum team members have at least familiar with the initial project features and has positive attitude in adapting Agile. Team member shall be expert in their area which is related to the project scope.
5. Scrum team members must be focused to the assigned project. If possible, the team members do not involve with other project.
6. Project duration can be chunked into small iterations where each iteration can produced workable solution to be showed to the product owner.
7. At minimal, the product owners has knowledge about Agile.
8. Implementation with internal projects where it is easy for the development team to verify the scope with the product owner.

Based on the observation, several types of projects are not suitable for Agile implementation, such as component and platform projects. This is because it is too insignificant to chuck the requirement implementation into small iterations. As for the component project, the features within the scope are quite small and it is not visible to separate the development into few sprints. Besides that those component is also interrelated. Thus it is not suitable to show the end result within sprint duration.

External project may also not suitable to implement Agile since it involves other organization resources. Besides that the team is also still familiarizing and adopting the Agile methodology.

C. Agile Requirement Development

Requirements development is an activity of understanding, eliciting and analyzing customers' expectations. This is the phase where the development team sits together with the product owners to identify their needs by listing the features required for the system. Team collects as much as information from the product owners to understand their needs. From the information gathered, they will further analyze and produce the user requirement.

In Agile, the product owners start with establishing the vision of the system. The vision statement describes the purpose of the system to be developed and what is the final system expectation [2]. Typically, vision statement consists of who the customers are, what customers need, and how these needs will be met. It portrays the essence of the system to be developed.

"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation" [3]. According to Agile Manifesto principles, people believe that communication is very important. Frequent discussions with the product owners will help the team to be aware of the expectation of the system developed. Agile illustrates the requirement in the form of user story.

User stories describe functionality that will be valuable either to a user or purchaser of a system or software [4]. User stories are composed of three aspects:

1. A written description of a story used for planning and as a reminder
2. Conversation about the story that serve to flesh out about the detail of the story.
3. Test that convey and document and that can be used to determined when a user story is complete.

User story description was captured with the basic information including – "As a *<role>,* I can/ want *<something/ feature>,* so that *<benefit/ reason>*". In other word, user story is a brief description that captures both the business and user requirements. It is a high-level definition of a requirement, containing just enough information so that the developers can produce a reasonable estimation of the effort to implement it.

There are two areas where user stories affect the planning process on Agile project. First is the estimation planning. From the list of user stories, the team will further estimate the effort in order to complete the task. It also works the other way around where, when it has been estimated, it can also help the team to breakdown the user story. It will then nicely complete within the defined iteration. Secondly is the schedule. Once the estimation has been done, team proceeds with the overall schedule. The list of user stories is captured in product backlog. The team then decides which user story to implement in which sprint. Therefore, a good user stories should be small, independent, sizeable and estimatible.

The other purpose of user stories is also for the testing team to develop test cases. The acceptance criteria should be discussed in the user stories with the tester. Due to these, more focus and effort need to be given to user stories development.

D. Product/ Sprint Backlog

A group of user stories is captured in list so-called, product backlog. Product backlog is a high-level list of functionality or requirement gathered to develop the solution/ product. In Agile perspective, it is a list of items that needs to be delivered by the end of the project. This was prepared at the initial stage of the product. Sometimes, a brainstorm session was conducted to identify all the features to be included at the end of the project.

Based on the list of requirements in the product backlog captured, the product owner shall be able to prioritize which features that he/ she wants to have it first within the first few weeks the project starts. Team identifies which user story will be executed in which sprint.

Prior to each sprint execution, the list of user stories were identified. This is called the sprint backlog which has the specific list of user stories together with the list of task applicable in that sprint. Those are the committed features to be delivered at the end of the sprint by the scrum team.

E. *Differences between User Stories and Use Case*

User story is one of the key Agile artifacts. It describes the requirement based on users' perspective. It is basically a simple story describing the ideas on what user wants the system to be developed. Story allows user to freely thinks and express their idea without any influence or worry on how the system will behave.

Use case is also one of the methodologies used in developing requirement. It describes the specific interaction between user and the system. In use cases, user is referred as Actor, where it is a role played by a person when interacting with the system. The elaboration is more on the system communicating with the actor. It details the step by step sequence between actor connections with the system.

In brief, below is the difference between User Stories and Use Cases [5].

TABLE I: Comparison between User Stories and Use Cases

User Story	Use Case
Focus on customer value. It is about user needs	Perspective of the users and their interaction with the system. It is about the behavior of the system
Emphasize the collaboration between the stakeholders and the team	Show the capabilities of the user and how these capabilities are met via a system response
As a … I want … so that …	Describes in steps on how the feature works
Written in conversational tone. Friendlier	Describes what action the user takes and how the system should responds
Written to facilitate release and iteration planning	Document an agreement between the customer and the development team

III. ADVANTAGES AND DISADVANTAGES USING TOOLS

In order to assist in Agile Implementation, MIMOS has provides tools to support the implementation. The tools given are Redmine and JIRA.

A. *Using Redmine*

Redmine is known as a project management web application which provides simple issue tracking feature [6].

Redmine tool was introduced during the initial Agile implementation in MIMOS. The tool is used to help the team capture the requirement. It lists all the product backlogs. Throughout the project, the scrum master monitors the team by breaking down the task according to the user story. Generally, for each user story description, the tasks were broken down into user story analysis, code development, unit test and functional test.

During the implementation, user stories concept were slowly introduced as to describe the minimal requirement of the features. With the help of Redmine, the team starts using the tool to capture the requirement in using user story concept. Previously, the team was more familiar with use case description.

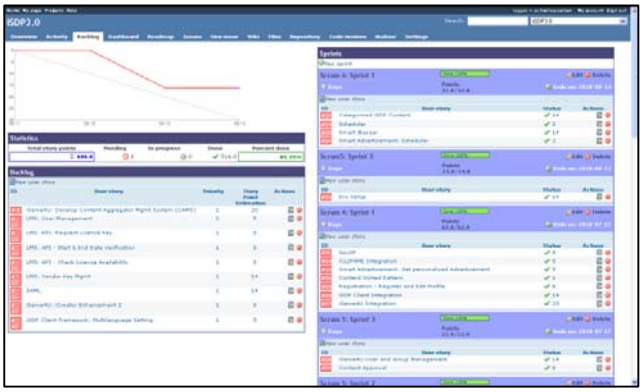

Figure 1: Redmine Project Dashboard

Figure 1 show how the product backlog has been arranged into sprint backlog in Redmine. The project has been divided into several sprints. In each sprint, the team has been assigned to develop few user stories.

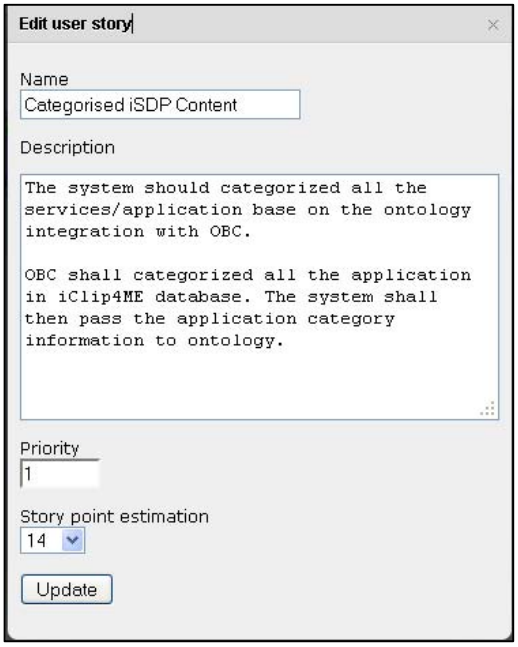

Figure 2: Redmine User Story Screen

Using Redmine, the user story elaboration consists of the name, description, priority and story point estimation. The user story name which will be the reference to what features is being called. The description box is to describe briefly the user story description. It explains who, what and why the feature should behave. These user stories, written by the business analyst, described based on the agreement between developers and testers therefore everyone could understands the purpose of

the requirement and value to the business. Refer to Figure 2 for the sample of user story elicitation in Redmine.

Once the user story has been elaborated, the user story is discussed and baseline. At that time, the product owners only involve at the higher level. They did not sit together during the user story elaboration. The team member decides the detailed feature. At the sprint execution, the business analyst together with the developer discussed the features for each user story in details.

Redmine was not used for bug tracking. Instead, bugs found during the project execution were not captured in Redmine. Bug tracking was recorded in different tool. Thus, Redmine was only use as to capture the requirement of the system by replacing the preparation of the Requirement Book.

B. Using JIRA

JIRA is widely used for issue tracking and project management [7].

Together with the organization initiative, JIRA has been evaluated further to replace Redmine. Usage of JIRA was used initially used to capture issues and tracking it. With the implementation of Agile in the organization, JIRA usage was extended to capture user stories, tracking the task progress for each user stories identified. Besides capturing product backlog, user stories and breakdown of task monitoring, JIRA is also been used for issue tracking.

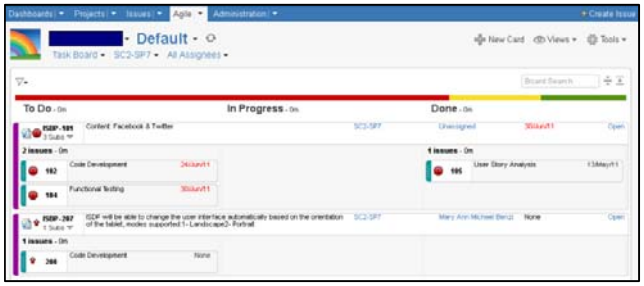

Figure 3: JIRA Task board View

JIRA provides several views such as Task board view as shown in Figure 3. This is basically a to-do list progress work. It is divided into three columns i.e. To Do, In Progress and Done. The list can be sorted according to the team member's view. Team members can drag-and-drop their task within the 3 columns. With this view, the team member can easily manage their own task. They will know what task has been assigned to them. The In Progress column indicates that he is currently working on a specific task. While the Done column shows that the task has been completed. Thus, the scrum master has a clear picture on the team's progress.

Besides Task board view, there is also Planning board view where it is suitable for the scrum master to have a general overview on the list of user stories and respective task per sprint view. The scrum master can easily assign the task to the respective team member.

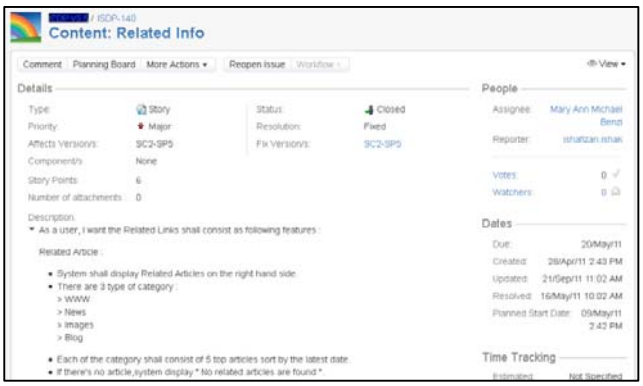

Figure 4: JIRA user story description

During this implementation, user story description has been improved by including the acceptance criteria. Acceptance criteria is written in a simple statement that expresses what the customer expects to see and can be used by the team to make sure the requirement is complete, or "done". Using JIRA, user stories are elaborated in more detailed. During this time, a simple guide has been prepared so that business analyst knows how to write the user story. The elaboration is divided into two parts. The first part provides the description of how the system should behave. The second part is the acceptance test where the additional notes used to determine when the user story is complete. It defines the boundaries of a user story, and is used to confirm when a story is completed and working as intended. Figure 4 shows the sample of how to user story is written in JIRA.

C. Redmine vs. JIRA

Based on the implementation feedbacks, below are the summarized comparison between Redmine and JIRA. It is grouped by the list of tool features.

TABLE II: Comparison between Redmine and JIRA tools

List of Features/ Areas	Redmine	JIRA
General	Manage issues and project managements	Manage features, list of tasks, improvements and bug tracking
Features Management	Able to capture list of features	Able to capture list of features
User story elaboration	Description of the user story space is limited. Attachment is put in centralized section	Description of the user story can be extended and edited. Attachments can be uploaded at the user story/ task created as the supporting document. Thus, easier reference to

List of Features/ Areas	Redmine	JIRA
		additional notes or attachment.
	Team members can't comment on issues related to the user story	User can put comments on the issues
User story card view	Non-customizable	Customizable
Bugs	During the implementation, bugs were recorded in different tools i.e. ClearQuest by IBM.	Bugs reported within JIRA tool and can be linked to specific user story. User knows the error relates to which bugs. JIRA has classification bugs as one of the issue type.
Estimation	Estimated time for issues has poor functionality. Estimation is only recorded as in hours.	Flexibility in writing the estimated effort. Allow to write in 3w 4d 12h format.
Task breakdown	Task breakdown can be listed for each user story	Task breakdown can be listed for each user story
Time tracking	It only adds on the total time spent.	Based on the estimation, it will also calculate the remaining effort required or expected.
History log	The changes of the user story description were overwritten. Once the content updated, the previous data is not kept.	The changes of the user story description are kept. The history log consists of list of edited Field, Original Value, New Value, edited by, date and time.
Agile board view	Has different type of views i.e.: 1. Backlog view – consists of list of product backlog, sprint backlog, and simple burndown chart 2. Dashboard view – workflow view. Easily drag and drop 'To do', 'In progress' and 'Done' state.	Has different type of views i.e.: 1. Planning board – manages development issues and task. 2. Task board – workflow view. Easily drag and drop 'To do', 'In progress' and 'Done' state. 3. Chart board – snapshot of the status of the project 4. Release board
Integration with AD	Able to integrate with organization AD	Able to integrate with organization AD
Integration with email	Able to integrate with organization email	Able to integrate with organization email. The email notification can be customized.

IV. Benefits on Agile implementation

A. *Benefits*

Agile focus on incremental and iterative development where requirements, design, implementation, and testing continues throughout the project lifecycle. At the start of the project, only the high-level features were identified. In each sprint, specified user story will be elaborated.

Traditional methodology is phase-wise manner, in which all of the requirements are gathered at the beginning, all of the design is completed next, and the development precedes once the requirement and design has been completed. The requirements are captured upfront with the customers. During that period, the entire requirement has been discussed, listed and agreed. With traditional methodology, most people assume that the requirements can be accurately gathered at the beginning of the project.

In Agile, sprint release/ demo were introduced to show the product owners on the workable solution. Product owners are being called for a short demonstration on the solution. At this stage, the features developed during the sprint has been completed and running well. The workable features will be shown. With the sprint release, the product owners shall provide the early acceptance as well as any related feedbacks. Those feedbacks will then be discussed and inserted in the product backlog if it is required to be implemented. This will help the team to deliver a better solution to the product owners as their involvement is throughout the project rather than having to wait until the end of the project to provide feedbacks or comments.

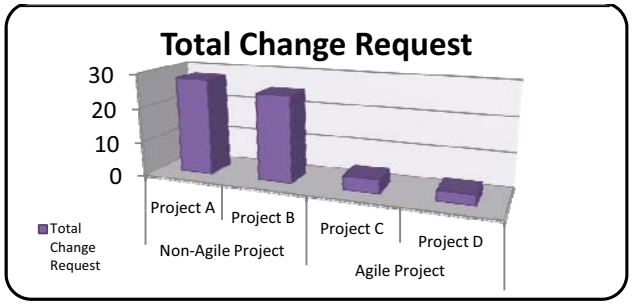

Figure 5: Comparison between non-Agile project and Agile project

With the newly introduced sprint release/ demo, the total number of change request raised during the project has reduced as shown in Figure 5 above. Figure 5 shows that change request in a project that adopt Agile methodology is approximately 85% less than project that adopt traditional methodology. Both the non-Agile project (project A and B) follows traditional v-model approach. Requirements were captured earlier, reviewed, baselined and signed-off. Code development starts and the test cases were also created after the requirement document has been baselined. During code development and testing against the requirement, team notices few changes needs to be incorporated. At least 25% of the change requests were originated from the requirement phase.

Comparing with the Agile project (project C and D), the development is done iteratively incremental. At the end of few sprints, the requirements were validated with the product owners/ stakeholders. During the review or demonstration, team gets better idea in developing the system. And, if there are any changes to be done, it was detected at the earlier stage rather than at the end of the system development. Besides that, with the demonstration, the team can easily get buy-in from the product owner. They have the clearer visibility on what is the end-result of the solution developed. From that, they are able to contribute more in the subsequent sprint during the user story elaboration.

B. Lesson learnt

Based on a few of Agile implementations feedbacks in MIMOS, most team agrees that daily meeting is a good discipline that they have adopted. Team members easily adopt to the routine of having a short meeting before starts the work for the day. During the meeting, team members are aware on each person's role in the project. Besides that, with the list of sprint backlog, the team members also felt more focused on the roles and responsibilities. They are aware with the features that they have to work on during the specified duration.

The usage of tools is important to assist in daily activities. The tools have made the team to have easy access to the list of the features of the solution. They are able to refer to the user story descriptions and also provide any additional inputs. They can also leverage on the collaboration feature within the tools, where they are able to communicate through the tools and discussion on the features to be developed.

The support from the management is also important in order to make the transition to Agile successful. Currently, in MIMOS Agile training is conducted to the technical team. However, all stakeholders should have clear understanding on Agile methodology. The support from them is important in order to make the transition to Agile successful.

V. CONCLUSION

By adopting Agile Methodology in the organization MIMOS quality of the project has improved especially in requirement phase. Requirement changes have reduced tremendously and managing requirements become easier with the assistance of tools such as Redmine and JIRA.

Due to the benefits above, MIMOS plan to implement Agile methodology and the usage of JIRA tool to the more challenging environment such as to MIMOS external projects where involves external parties such as other ministries and government agencies. MIMOS also intend to implement to the project where project team members are geographical distributed with the intention to control requirement changes hence projects able to complete within the time and cost.

REFERENCES

[1] Mike Cottmeyer, V. Lee Henson, VersionOne whitepaper "The Agile Business Analyst"

[2] Schwaber, Ken., Agile Project Management with Scrum. Microsoft Press. 2004, page 68

[3] Principles behind the Agile Manifesto: "http://agilemanifesto.org/principles.html"

[4] M. Cohn, "User Stories Applied: For Agile Software Development", 2004

[5] Andrew Stellman., May 3, 2009, Available: "http://www.stellman-greene.com/2009/05/03/requirements-101-user-stories-vs-use-cases/"

[6] Redmine Available: "http://www.redmine.org/"

[7] JIRA Available: "http://www.atlassian.com/software/jira/"