



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

A Pragmatic Study of Tools Used in Agile Software Development

Sanjeev Dhawan^a, Divendu Kumar Mishra^b

^aFaculty of Computer Science & Engineering, ^bResearch Scholar

Department of Computer Science & Engineering, University Institute of Engineering and Technology,
Kurukshetra University, Kurukshetra-136 119, Haryana, INDIA.

E-mail: rsdhawan@rediffmail.com, raton01mishra@gmail.com

Abstract: Agile methodologies are gaining popularity in industry although they compromise a mix of accepted and controversial software engineering practices. A number of tools are available to manage a project with Agile software development but it is a very difficult task to select a tools or techniques that fully support your software development work. Some tools support distributed Agile software development. With distributed Agile development it is possible to tap into new global markets and make best use of globally available talent, while potentially reducing cost. It is always beneficial to provide distributed teams with the right tools to work as effectively as possible and remove as many of the barriers created by distribution as possible. Therefore, in this paper an attempt has been made to discuss the comparison of some of the tools that is widely used in the industry.

Key words: Rally, VersionOne, Sprint, Sprint Planning.

I. Introduction

Agile software development adoption spreads over the industry. Within the last 10 years, Agile methods jumped from nowhere to the peak. Different tools and platforms were developed for the Agile development. Some of them are online tools along with off-line also. Some tools support distributed team and some of them best work with collocated team. In 2008, a state of Agile Development survey was conducted by VersionOne. In this survey 57% of respondents stated that their teams were distributed [1]. Thereafter in same survey held in 2011, 65% of respondent stated that their teams were distributed [2]. Furthermore in 2008, 41% of respondents said that they were currently using, or plan to combine, Agile with outsourced development. While collocating the team, it is clearly recommended approach that many teams are unable to do this and are faced with trying to stick to their Agile principles and apply Agile practices to distributed development. This paper describes some of the proven practices about distributed Agile software development and the comparison between tools used in distributed Agile software development.

II. Principles Behind the Agile Software Development

The Agile Alliance also documented the principles they follow that underlie their manifesto [3]. As such the Agile methods are principle-based, rather than rule-based. Agile software development consists of predefined rules regarding the roles, relationships, and activities, the team and manager are guided by these principles:

- a. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- b. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- c. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time scale.
- d. Business people and developers must work together daily through the project.
- e. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- f. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- g. Working software is the primary measure of progress.
- h. Agile processes promote sustainable development.
- i. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

- j. Continuous attention to technical excellence and good design enhances agility.
- k. Simplicity – the art of maximizing the amount of work not done – is essential.
- l. The best architectures, requirements, and designs emerge from self-organizing teams.
- m. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

III. Agile Methodology Used

There are various methodology used in Agile development like Extreme Programming (Xp), Scrum, Crystal, Feature Driven Development (FDD), Kanban and Scrumban. Scrum or scrum variants continue to make up more than two-third of the methodologies being used. In 2011 state of Agile Development survey was conducted by VersionOne [2] which concluded that mostly used methodology for Agile software development is Scrum. Most of the tools used in Agile Development follow the Scrum methodology. Table 1 shows the use of Agile methodology.

Table 1- Use of Agile Methodology

Agile Methodology	Use of Agile Methodology (in %)
Scrum	52
Scrum/Xp Hybrid	14
Custom Hybrid	9
Kanban	3
Scrumban	3
FDD	2
XP	2
Lean	2
Agile Unified Process	1
Agile Modeling	1
Dynamic System Development	1
Other	10

IV. Tools Used in Agile Methodology

According to Gartner [6], globalization of software development has expanded rapidly in recent years and has brought in its wake changes that impact application development projects. Global Software Development (GSD) is becoming the norm for many technology companies. The most common type of tools currently being used (or planned) are: Automated Acceptance Tools (20%), Release Management Tools (16%) and Continuous Integration Tools (16%). Interestingly, the Agile tools being used the least today is also the most desired tool of tomorrow – Ideas Management Tool [2]. In terms of specific Agile software tools, most using standard office productivity office productivity tools such as excel, followed by specialized tools like VersionOne. The use of bug trackers has come on strong this year (14%). In the next part of this paper, the working of mostly used tools VersionOne and Rally is discussed.

V. Working with VersionOne

VersionOne provides software and tools for Agile project management, enabling today's most popular Agile methodologies: Scrum, XP, Lean/Kanban and Hybrid. All VersionOne's products are designed from the ground up to support today's leading-edge development methodologies. From small teams to distributed enterprises, VersionOne's suite of Agile project and life cycle management tools helps companies scale Agile faster, easier, and smarter. The general Agile process for planning and tracking is mirrored in the application's product and planning section. The product planning allows the customer to define, organize and prepare the project team's backlog. New stories are created to define new functionality to be added to the system. Some of the major function of product planning include: Enter new backlog, Detail backlog through descriptions, Break down large epics into smaller epics, and Estimate the backlog with the help of the team. The product and planning consists of the following:

- a) Release Planning: Release Planning is an optional step in the process to allow the team to carve off a portion of the backlog for delivery at a certain point. Teams who deliver to production after each sprint, on the other hand, may have no use for releases at all. Major functions of release planning is to define new

releases as sub-projects in the project tree, Schedule items into releases; forecast the delivery point of a release based on historical data, and to define regression test plans.

- b) **Sprint Planning:** The sprint represents the basic timeframe for an Agile team. Within a given sprint, the team is fully delivering a small slice of the overall feature set. It includes the design, development, testing and documentation of the features. Major functions in sprint planning consist of scheduling the sprint work and breaking down to more detailed one.
- c) **Sprint Tracking:** Once the sprint is planned out by the team, they turn their attention to execution. As team members work on items, optionally track time, and record progress, the tracking pages are used to keep their teammates and stakeholders in the loop and sign up for additional work when necessary. Some of the major functions in sprint planning include: Sign up for work, Update status, and Track time and close items.

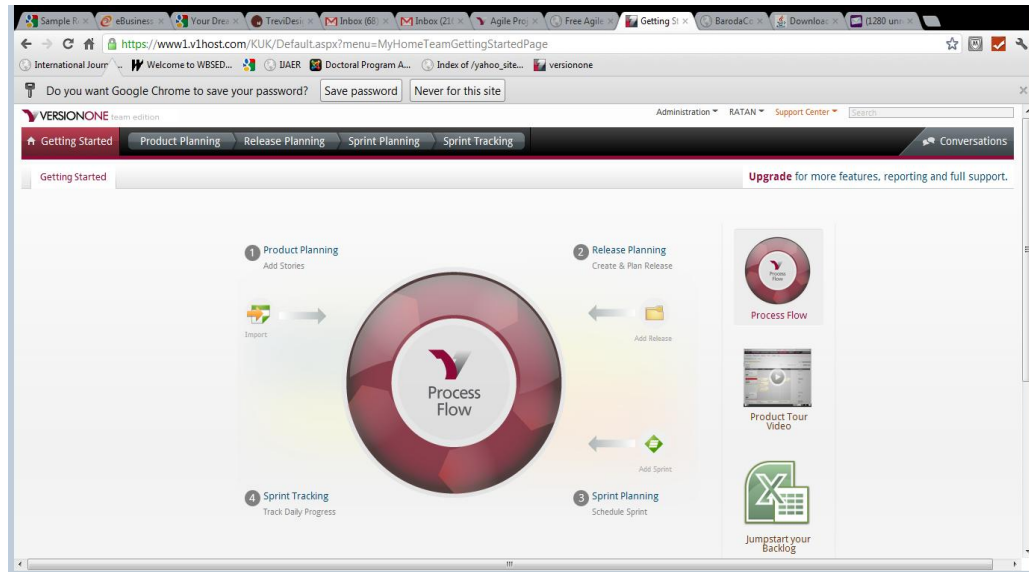


Figure 1 - Basic Planning Menus of VersionOne.

VI. Working with Rally

“Rally” is one of the tools for Agile project management. Rally provides software and tools for Agile project management, enabling today’s most popular Agile methodologies: Scrum, XP, Lean/Kanban and Hybrid. All Rally’s products are designed from the ground up to support today’s leading-edge development methodologies. From small teams to distributed enterprises, Rally’s suite of Agile project and life cycle management tools helps companies scale Agile faster, easier, and smarter. Rally is used to coordinate, plan and track software development project plans, priorities and progress. The general Agile process for planning and tracking is mirrored in the application's planning menu:

- I. **Plan:** It consists of five drop down menus in Plan named as Backlog, User stories, Release, Iteration, and Plan.
 - (a) *Backlog:* The backlog is the collection of all customer input represented by user stories, and any open defects or defect suites.
 - (b) *User Story:* A user story is a statement of user functionality formulated as a few sentences in the everyday language of the user that can be completed within iteration.
 - (c) *Release:* A release is a milestone representing the internal or external delivery of a working, tested version of the software system under development.
 - (d) *Iteration:* Iterations are short development cycles that produce working, fully-tested software.
 - (d) *Plan:* Specify the release you are planning from the Release drop-down, which displays all active or planned releases.
- II. **Track:** In the tracking phase the main task is to track the iteration task status, team status and release task status.
 - (a) *Iteration Task Status:* This view provides a colored iteration Indicator box displays critical iteration information.

- (b) *Team Status*: The Team Status view shows all project team members in bold font, with their assigned tasks and individual capacity by project for the selected iteration.
 - (c) *Release Task Status*: Release Status app is a management view of work across workspaces that share the same release schedule. A shared release schedule is determined by a release with the same name, start and end date.
- III. Quality**: The role for the Quality Professional often takes center stage. These guides leverage Rally's real world experience to help you quickly grasp the basic practices of Agile Quality.
- IV. Report**: Rally provides powerful reporting capabilities to track software team's progress. Reports and metrics can be easily shared with all stakeholders using email and common electronic formats.

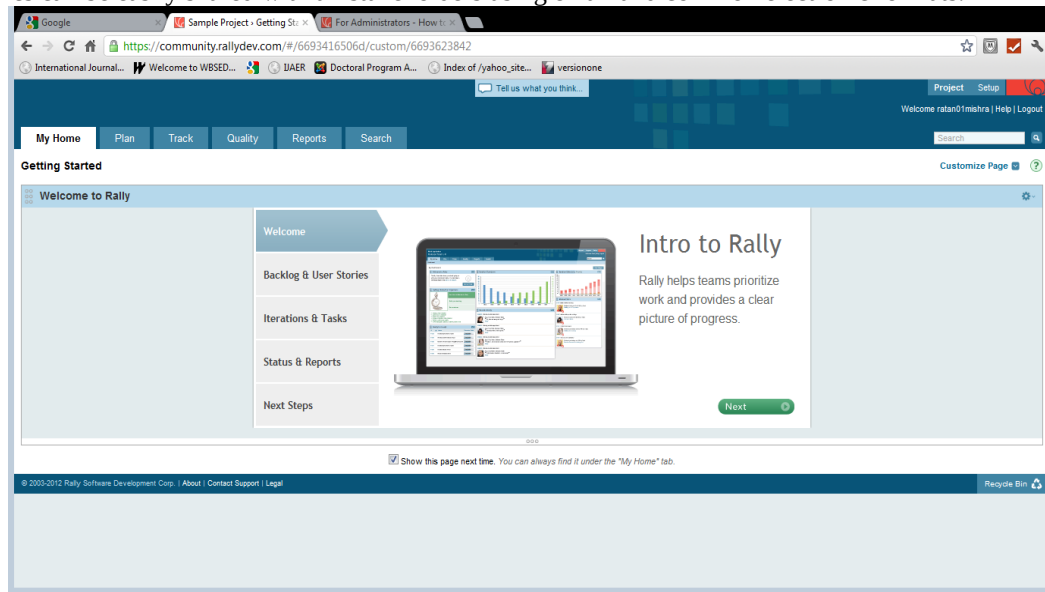


Figure 2 - Basic Planning Menus of Rally.

VII. Conclusions

Both companies did extensive demos for their leadership team and key stakeholders. Both tools intrinsically are built around the scrum Agile process. Both were priced around the same level with VersionOne being just a little less per seat, per month, but Rally matched and beat this price point in negotiations. The huge gaping hole in VersionOne is that it really didn't assist with resource planning at all. That is, they don't enable you to enter the amount of available resources in terms of hours, days, etc, and then in the planning cycle show you where you are in using those hours as you take stories from the backlog and add them to the sprint. Both tools track burn down during the sprint of course, but only Rally lets you know if you are planning too many stories in the sprint. Even XPlanner supports this, so it is a big miss for VersionOne. Therefore, we can only assume that they are working to add this capability. Also, the rollout reporting for an entire release is more powerful and flexible in Rally. This was a big plus for Rally. To be sure, Rally isn't super sophisticated in resource planning. It doesn't allow the individual team members to enter their availability and then sum it up for the sprint. So there is an urgent need to add this feature to the Rally Community. It will allow you to add the total number of hours available for a sprint at the beginning of the planning cycle. So, it is up to the developers to figure out the possible solutions for total number of hours available and remaining for the project panning and implementation.

VIII. References

- [1] 3rd Annual Survey: 2008 "The State of Agile Development" http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf.
- [2] 6th Annual Survey: 2011 "The State of Agile Development" http://www.versionone.com/state_of_Agile_development_survey/11/.
- [3] F. P. Brooks, The Mythical Man-Month, Anniversary Edition: Addison-Wesley Publishing Company, 1995.
- [4] <http://www.versionone.com/>
- [5] <https://community.rallydev.com>
- [6] Iyengar, P., Application Development Is More Global than Ever, Publication G00124025, Gartner, 2004; www.gartner.com/resources/124000/124025/application_dev.pdf.