A Comparative Study on Cloud-based Agile Tools

Mohsen Taheri and S. Masoud Sadjadi
School of Computing and Information Sciences
Florida International University
Miami, FL 33199, USA
mtahe006@fiu.edu, sadjadi@cs.fiu.edu

Abstract

In the context of software development, companies, organizations and developer teams want to develop pure software products more efficiently and quickly. To deal with the new issues which accompany growing projects and software product complexities, agile tools boost simplicity and accelerate team’s collaboration in a single framework. Adoption of agile tools can be a difficult process, due to agile tools must match company requirements to enhance the success of software development projects. The purpose of this paper is to introduce key criteria in details to consider in cloud services and cloud-based agile tool selection; also it presents top agile tools comparative classification based on a practical research to satisfy software development requirements under a cloud perspective.

Keywords: software development; agile tools; cloud-based tools; comparative classification.

1 Introduction

In the last decade, agile software development has been popular in software industry due to its iterative and dynamic nature. The flexible nature of agile software development (ASD) supports projects with high instability in requirements analysis. ASD is a set of software development methodologies based on incremental and iterative development in which specifications and alternatives develop by means of cooperation between self-organizing, cross-functional groups. It promotes adaptive arranging, evolutionary improvement, early delivery, ongoing enhancement, and encourages rapid and accommodating response to change. Also, some other benefits obtained from implementing agile consist of the ability to deal with the software development visibility, cost, risk and priority management, to improve team moral and to make simpler project implementation process. This research is based on standard agile definitions and concepts and uses agile principles and agile manifesto to review the tools and their features. According to the agile manifesto, “individuals and interactions are over processes and tools, working software is over comprehensive documentation, customer collaboration is over contract negotiation and responding to change is over following a plan” [7] (book).

The companies’ products and services with respect to agile software development and project management tools is now becoming more mature with simple or complicated commercial tools and dozens of different vendors which guide you to learn and work with agile methodologies. This paper tries to present how a special agile tool fully meet all company expectations as a “one size fits all” tool for a product team, and make their collaboration and project tracking overall enjoyable.

The selection problem appears quite simple, but without analysis the clients realize how complex it really is. The necessity for continuous interaction and collaboration among team members and having access to specified resources demands pervasive service. Therefore, to be able to satisfy the collaboration and requirements of agile teams, cloud tools and cloud computing solutions are used. Additionally, agile approaches cope with remarkably unpredictable needs accompanied by rapid progress cycles and frequent customer feedback that involves plenty of cooperation and resource sharing. Cloud solutions make it easy for agile methods to face constant adjusting demands through avoiding considerable communication gaps. Although there are many apps and tools offering traditional project management, tasks management & To-Do List planning, this paper focuses only on agile project management tools, their specification and a classification to select appropriate agile tool in regard to cloud requirements.

In this paper after a comprehensive comparative classification and a feature-based taxonomy a case study regarding usage of cloud-based agile tools is presented. The studies show advantages of using cloud-based solutions accompanied with agile methods. This paper is organized as follows: Section 2 presents the literature review and previous related works. Section 3 presents the cloud-based ASD. In section 4, the methodology and the research steps conducted during the study is provided and also it describes the criteria used for evaluating currently existing tools. Section 5 explains findings and comparison table and Section 6 presents a case study. Finally, Section 7 concludes with final remarks [20] (proceedings), [15, 4] (book).
2 Related Works

The present and existing white papers, journal and conference papers and best tool usage surveys in the agile development context are impotent sources in this paper. We went through many most important world’s largest scientific and educational sources such as IEEE, ACM, Springer, Google scholar, and etc. and even surfed through less scientific online sources such as websites, whitepapers and published surveys. Finally, few different surveys, which some of them were sponsored by tool vendors themselves were found. Some of the most relevant works to this research are presented as follows.

In 2011 and 2012 Azizyan [18, 5] (proceedings) presents a group of characteristics that are most desired by the existing software companies which shows that the most satisfactory tool attribute is ease of use. This paper helps us to prepare a list of tools, criteria and metrics for our tables. Also Azizyan presents a journey towards agile tool selection for a specific anonymous company and the tool selection process is based on a study of the tool no functional features such as flexibility and usability. This paper gives a brief description of the company, then another section lists and describes the metrics used for evaluating currently existing tools. It has focused just on a special company and few tools, but in comparison with other papers, it introduces a methodology to select the right tool.

[6] (online) focused on collecting statistics on tools used in requirements management, and also there are some statistics on agile method used and reasons for selecting an agile project management tool in 2006. It helps us to prepare a list of tools, criteria and metrics for our tables.

In 2008 [8] (online) mainly focused on tools used in agile projects. It focused on gathering statistics on company structure and maturity of agile methods using TargetProcess trial versions. Although the paper has published a couple years ago and in recent years, many new tools have captured the market, it is beneficial as a reference to choose most important tools and metrics.

In 2013, [23] (online) written by the VersionOne Company includes a normalized and wide distribution of responses of multitude of channels from companies, engineers, scrum masters, product owners and even self-employed engineers. The respondents are from different countries and questions have focused on details such as reasons for adopting Agile, agile techniques used. The main points of the paper are detailed statistics in the agile methods in projects, and the information about adopting agile methods.

3 Cloud-based ASD

Agile software development comprises a set of several iterative sprints with high collaboration among distributed teams and collaboration have high priority rather than processes. Scrum is one of the agile software project management methods which in sprints are the basic units and the team members work on several product features and each tem have high degree of self-management.

Cloud computing provides an easy access to resources to facilitate the reusability and are categorized into: Software as a service (SaaS), Infrastructure as a service (IaaS) and Platform as a service (PaaS). To adopt agile software development methods, during the product lifecycle, there are many collaboration demands between customers and developer teams. These interactions can use efficiently by cloud computing and cloud services. In addition, cloud computing optimizes functionality and cost flexibility by using on-demand and metered services or Pay-Per-Use resources [3, 22] (proceeding).

In the concept of cloud agile software development some benefits are more highlighted. Cloud services enhance productivity per project or artifact, improve convenient access to data anywhere, anytime, and decrease major costs for software and hardware. Also cloud make flexibility in scaling without serious financial or people issues. Agile tools using cloud environment improve collaboration and communication among distributed teams and available infrastructure by supporting rapid development increase delivery on artifacts. Figure 1 shows how agile and cloud may interact to satisfy company’s software development needs [16] (proceeding).

![Figure 1: Cloud-based agile software development](image)

4 Analysis

According to the concepts of agile software development we know that "Individuals and Interactions" is more essential than "Processes and Tools"; but the right agile tools really can affect the enterprise, especially when interactions can be more productive. How top agile tools are listed and which
main factors are considered are going to be discussed in this section.

4.1 Cloud-based agile tools

In this part two evaluation methods are implemented and to list the top tools in the market, many blogs, web pages, reviews, tutorials have been read. Afterward, we reviewed papers, surveys, and white papers, especially those which had been published in recent years. In the next phase, more than 40 graduate students, including 10 PhD students from the computer science department of Florida International University (FIU) during a semester have studied agile tools and have done practical research in the company or project and finally they made a video describing tool, their features and a tool usage in a project. The final analysis and evaluation are released and available on Github and YouTube [24] (online).

In our selection in this phase, we considered some features like the size of the project, the size of the team, the stability of the requirements and the complexity of the software for a wide range of available tools to maintain diversity among them. [11] (book).

4.2 Essential factors

In this phase all of the recent surveys were studied. In addition, some feedbacks provided by students helped us to balance some of the vendor’s surveys. Five different core criteria definition are presented as follows [9] (book), [12] (journal). During the semester each student after studying one or two tools, he/she was supposed to present and review a comprehensive slide show and hands-on practice in the class

1) Responsiveness
   Responsiveness is how the vendors respond to the needs of their customers. How do the vendors support their customers?

2) Ease of Use
   How users can utilize the agile tool without a lot of training and time consuming procedures.

3) Category
   Companies are placed into the categories that fit with their organization’s needs. For example, if it is an organization with a hundred users, it is probably not going to want a simple standalone solution.

4) Flexibility
   The agile project management tool should have flexibility to adapt to differences in fundamental company’s requirement.

5) Features
   Which specifications and features a system supports, and how those features would be used to perform the project process [10] (book).

5 Classification and Results

After evaluation in the previous part, the comparison tables “Table 1, 2” have been created [20] (proceedings). The purpose of the comparison table is to highlight the requirements and to be able to compare different agile tools through specified requirements.

<table>
<thead>
<tr>
<th>Deployment, Integrity and Security</th>
<th>Web services API; Project-level security; Free trial software available; Maturity size-based product versions; Integrates with Existing Tools like Source control systems (e.g. GitHub), bug trackers (e.g. JIRA).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workspace and Process</td>
<td>Extensive options for boards, fields; Drag and drop story, task and boards; Customizable methodologies (XP, Scrum, Kanban, etc.); Color coded visual representation.</td>
</tr>
<tr>
<td>Collaboration</td>
<td>Reporting and tracking for distributed team members; Communication media for teams; Mobile Stream to keep projects moving; Email notifications and RSS feeds; Customizable boards and coding.</td>
</tr>
<tr>
<td>Analytics, Visibility and Reporting</td>
<td>Advanced planning e.g. what-if analysis; Dashboards with sufficient metrics; Reports, charts and graphs; Hierarchy charts, Relationship mapping, Release dependency visibility.</td>
</tr>
<tr>
<td>Simplicity &amp; Ease of Use</td>
<td>Shortcut options for actions such as: Close, open and delete; Customizable dashboards for tracking, Drag and Drop; Interactive environment supporting the daily activities of teams</td>
</tr>
<tr>
<td>Program Management</td>
<td>Epic planning; Cross- team planning, tracking; Release rollouts; Program-level Epicboards.</td>
</tr>
<tr>
<td>Lifecycle Coverage</td>
<td>Iteration planning and its tracking, Strategic Goals, backlog and the repository for defects, Test management; Product and their release.</td>
</tr>
</tbody>
</table>
Some of the most important key factors that should be considered to select an agile tool are as follows:

- Life Cycle Management using One Agile Tool

Storing project information in the different multiple tools leads to inaccurate results and prevents real-time visibility; therefore using one unique agile tool is able to facilitate clients to comfortable implementation.

- Cross-Functional Teams

It means to manage the requirements of the customers, programmers, testers, product owners, and other stakeholders in an integrated environment to enhance collaboration and consistency.

- Enterprise Scale

In order to deploy an enterprise, agile tools should be able to handle the project structure, tasks, defects and tests.

Table 2: Top Agile tools comparison chart (A: Full support, B: Quite good and suited for some purposes, C: Not appropriate)

<table>
<thead>
<tr>
<th>Lifecycle Coverage</th>
<th>Proprietary tools</th>
<th>Commercial Cloud-based Agile tools</th>
<th>Open source tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
</tr>
<tr>
<td>Analytics, and Reporting</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
</tr>
<tr>
<td>Program Management</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
</tr>
<tr>
<td>Deployment, Integrity and Security</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
</tr>
<tr>
<td>Scrum &amp; Kanban Supported</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
</tr>
<tr>
<td>Popularity on the web</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
<td>A A A A A A A A</td>
</tr>
</tbody>
</table>

The comparison tables focus only on top 22 agile project management tools and compare them; but there are also a lot of commercial vendors offering solutions in this market. Thus, how to select right agile project management tool for different projects? Agile only fits in some company scales and the sad truth is that agile doesn’t fit all company scales. So, many agile adoptions in progress right now are going to fail for this reason. A cloud-based simple solution and guide to this issue is presented in Figure 2 that shows how a client can select a tool in a systematic process.

6 Case study

As an evaluation for proposed classification, a case study based on our findings performed and 20 software projects including customized software projects, senior projects and startups with similar characteristics and same conditions in regard to size, complexity, time, human and technical resource performed. In next experiment during next semester another 20 projects according to the first experiment conditions performed but the traditional tools which were non-cloud provisioning had been applied. In brief some key factors in this case study replied to these considerable questions:

1. Flexibility: Can the system adapt to how your organization does business? 2. Ease of Use: Will your people be able to use the tool without a couple of hours training? 3. Category: Into which classification of agile project management tools does it fit, and does that class match with the needs of your organization? 4. Responsiveness: How responsive is the organization? 5. Pricing: Does the pricing of the system match the value you will receive? 6. Features: Does the system have enough features to meet your current and future objectives?

Then we classified them in a table based on comprehensive factors:

- Feature-driven Development: Some companies attempt to use a traditional tool that causes their project to be more complicated due to these tools don’t support basic agile practices [17] (book).
- Lifecycle Management: Storing project information in different multiple tools causes inaccurate results and prevents to comfortable real-time visibility [19] (book).

- Cross-Functionality: It means to manage the requirements of the customers, programmers, testers, product owners, and other stakeholders in an integrated environment to enhance collaboration and consistency [13] (journal).

- Configuration with Flexibility: An agile management tool should let companies to organize, and plan according to their requirements.

- Simplicity: Like agile software project development, the simple one with ease of use is better, but the level of maturity is considerable.

- Enterprise Scale: In order to deployment of an enterprise, agile tools should be able to handle the project structure, tasks, defects and tests [2] (book).

Finally findings analysis represented a remarkable gap in projects quality as shown in “Table III” [1, 14, 21] (journal).

<table>
<thead>
<tr>
<th></th>
<th>Cloud-based agile software development</th>
<th>Traditional and Non-Cloud-based agile software development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature-driven Development</td>
<td>90</td>
<td>75</td>
</tr>
<tr>
<td>Lifecycle Management</td>
<td>90</td>
<td>83</td>
</tr>
<tr>
<td>Cross-Functionality</td>
<td>95</td>
<td>70</td>
</tr>
<tr>
<td>Simplicity</td>
<td>98</td>
<td>77</td>
</tr>
<tr>
<td>Enterprise Scale</td>
<td>85</td>
<td>65</td>
</tr>
</tbody>
</table>

7 Conclusion

As project team members in the company continue to use agile and enterprise scales agile development within their companies, the challenges of managing different groups continue to increase. Agile software development tools provide solutions to manage this sophisticated process using a framework to maximize the consistency and success of agile development. In this paper, we presented a feature-based classification approach to select the right cloud-based agile tools. Finally, we classified results in a table and presented a solution and a case study to select right agile tool based on features of agile software development tool and enterprise needs.
ACKNOWLEDGMENT

This research was partly supported by the Leona M. and Harry B. Helmsley Charitable Trust via the Georgia Tech’s Vertically Integrated Projects (VIP) Program. The authors also thank the FIU graduate students attended Dr. Sadjadi’s Fall 2014 Advanced Software Engineering class for their technical assistance and insightful discussions during the class sessions. This material is based in part upon work supported by the National Science Foundation under Grant Numbers of I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-1429345, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, and MRI CNS-0959985.

REFERENCES

[24] https://www.youtube.com/channel/UCuclIZz2RRY6tYcKuTDxO8pw.