

Analysis on the Scale Agile Framework Approach

Bhavana Vakkalagadda[†]

Software Engineering
Arizona State University
Tempe Arizona USA
bvakkala@asu.edu

SaiRam Eadala

Software Engineering
Arizona State University
Tempe Arizona USA
veadala@asu.edu

ABSTRACT

Agile is one of the most used Software Development Methodologies used in the Software Industry currently. Agile manifesto promotes mainly the Iterative Development Team Collaboration and Change recognition. Short Iterations are used that are required for efficient Product Delivery. Traditional Methodologies does not support rapid changes in the development process. Despite the advantages of Agile, criticism on agile methodology states that it fails to pay attention to architectural and design issues. Here in the paper we will investigate one of the agile frameworks which helps us to overcome these issues which is known as SAF

Keywords- Agile, Agile Manifesto, Framework, Teams, Principles, Product Owner ,Scrum

I. INTRODUCTION

Agile Methodologies are basically iterative and incremental development approach. The four major characteristics of agile methodologies are: adaptive planning, iterative & evolutionary development, rapid and flexible response to change and promote communication [1, 2]. The main principles of Agile methodologies are Welcome Changes, Deliver Frequently and Organized Teams. As it is also called as lightweight process, it will be more suitable for the development of small projects. Agile software development takes the view that teams should start with simple and predictable estimate to the final requirement and then continue to increment the detail of these requirements throughout the development Life Cycle. The main deliverables of the Agile Project are Product Vision Statement, Product RoadMap, Product Backlog, Product Release plan, Sprint Backlog, Increment. These incremental requirements refinement changes the scope of the design, coding and testing at all stages of production activity. In this way, the requirements work product is as accurate and useful as the final software itself [4].

Agile methodology addresses exactly the challenges of an unpredictable, disordered business and technology environment [5]. Agile methodologies are used to achieve higher quality software in a shorter period of time, self-organizing teams, customer collaboration, less documentation and reduced time to market [6,7]. Agile methodology includes a family of lightweight methods that include Scrum, Crystal Clear, Extreme Programming (XP), Adaptive Software Development (ASD),

Feature Driven Development (FDD), and Dynamic Systems Development Method (DSDM) Crystal, Lean Software Development etc. [8]. Agile methods break tasks into small increments with minimal planning called Sprints. Sprints are short time frames that runs from one to four weeks. Each sprint involves a team working through a full software development cycle, including planning, requirements analysis, design, coding, unit testing, and acceptance testing. This minimizes overall risk and allows the project to adapt to changes quickly. Most of the agile implementations use a formal daily face-to-face meeting among team members. In this brief meeting, team members report to each other what they did the previous day, what they intend to do today, and what are the hurdles they faced [13, 14,15].

As per Bustamante and Sawhney (2011), the best Agile project team is little in terms people in the group, communicate face to face daily through standup call which has a team size of people not greater than 9. Beside this, agile ways like scrum master optimum team size of seven plus or minus two. But some business specialists claim that Agile may be a one size fits all methodology which it may be scaled up to a hundred- and fifty-person team. Normally Agile Management is long-faced with one key challenge: What should the optimal team size be for which Agile is to be applied for the team to be effective? Since Agile is individual focused, we ought to perceive how team size affects individual behavior and productivity at intervals in agile team and this rises to a different challenge. What people factors should be considered when managing an agile team to make it effective?[9,10]

II.SCALE AGILE FRAMEWORK

Scale Agile Framework contains teams of 5-9 Individuals who define, build, test and deliver an increment of value in a short time box. The main reason behind having two small teams rather than having one large team is the communication between the team diminishes if the team size increases more than required.[19]

These teams contain two main roles namely the Product Owner and the Scrum Master. Working with multiple teams over in agile method is known as Agile Release Train. These are responsible for delivering large solution values. The teams that are in this train collaborate with each other contribute towards the project vision.

They are also responsible for the Continuous Delivery Pipeline.[19]

Agile teams are the build blocks for creating and delivering value. Without the results will not be as expected. These teams should be self-organizing and self-managing teams that must be accountable to deliver the results that meets the expectations of customers and the stakeholders. They should also be accountable to one and another for deliver the product on time.[19] Teams in the Scale Agile framework have the following responsibilities to be taken care by them.[19]

1. Collaborate with the Product Owner to create and refine user stories and acceptance criteria
2. Participate in PI Planning and create Iteration plans and Team PI Objectives
3. Develop and commit to Team PI Objectives and iteration goals
4. Estimate the size and complexity of their work
5. Use pairing and other practices for frequent review
6. Determine the technical design in their area of concern, within the architectural guidelines
7. Conduct research, design, prototype, and other exploration activities
8. Implement and integrate changes in small batches
9. Create and test the work products defined by their features
10. Test the work products defined by their features
11. Deploy the work products to staging and production
12. Support operational business solutions
13. Support and/or create the automation necessary to build the continuous delivery pipeline
14. Continuously improve the team's process

III. DESIGN AND ARCHITECTURE

Architecture is an important aspect of agile software development method, just like in normal models like Waterfall Model, and is a critical part of scaling agile approaches to meet the real-world needs of modern organizations. Architecture which is well designed helps the customer to understand the product in the better way. Changes can be implemented in case if customer feels that he may have missed something during the sprint cycle. Agile architecture must strike the balance between the team, software and environment.[17]

In large Agile Teams, Geographically Distributed Teams there will be Architecture Owner Team or Enterprise Architecture team that will take care of these activities' Geographically Distributed teams architecture owner in each sub team will be in architecture team which helps a chance to increase the chance that each sub team understands and follow the architecture as well as increases the chance that the overall architecture strategy will address the full needs of the overall solution. One Chief Architecture owner will be there in the architecture team and it will be shifting from one person to another person within the team.[16]

There are two ways where we can implement agile software development where we have multiple teams collaborating with each other.Lets see the advantages and disadvantages for both of them and how these influence Design and architecture of the system.[18]

First approach is to have a few teams of developers and a separate team of software architects work closely with one another.

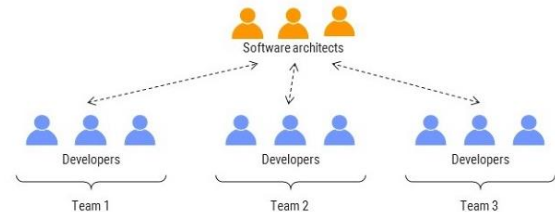


Figure1: Separate Team of Architects Works with Multiple Development Teams.[20]

We may face few problems while following the approach. The drawbacks to this approach are

1. In the above scenario the software architects will become the demanders who define the vision whereas developers are left to implement it. In case of noninvolvement of the developers in few of activities may cause a gap between them which may reflect on the final product[18]
2. Second approach would be placing a software architect in the same team as that of developers which helps in the proper communication between the software architect and developer of same team as well as software architect between different teams.[18]

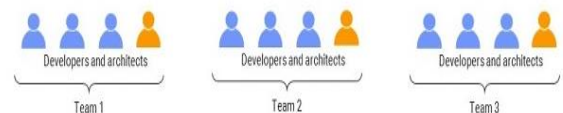


Figure2 : Each Development Team has One Software Architect[20]

This approach has advantages and responsibilities changes for the software architect. This approach helps us in making quick informed decisions by making the prototype of the system whether the functionality can be implemented or not. This approach focuses on the Sustainability where the architecture will support the project for a long term. In this approach the software architect is part of development team. He can give first hand decisions about the software developed by the development team.[18]

ACKNOWLEDGMENTS

I would like to thank my professor Michael Findler for encouraging us to do research on agile team working with multiple agile teams

REFERENCES

[References:

- [1] Andrew Begel, Nachiappan Nagappan, "Usage and Perceptions of Agile Software Development in an Industrial Context:An Exploratory Study", First International symposium on empirical software engineering and measurement, pp. 255-264, 2007.
- [2] Peter Maher, "Weaving Agile Software Development Techniques into a Traditional Computer Science Curriculum", Proc. of 6th IEEE International Conference on Information Technology: New Generation, pp. 1687-1688, 2009.
- [3] Anfan Zuo, Jing Yang, Xiaowen Chen, "Research of Agile Software Development Based on Formal Methods", International Conference on Multimedia Information Networking and Security, pp. 762-766, 2010.
- [4] Michael J Rees, "A Feasible User Story Tool for Agile Software Development", Proc. Of 9th Asia-Pacific Software Engineering Conference (APSEC' 02), 2002.
- [5] Richard Mordinyi, Eva Kuhn, Alexander Schatten, "Towards an Architectural Framework for Agile Software Development", 17th IEEE International Conference and workshops on Engineering of Computer Based Systems, pp. 276- 280, 2010.
- [6] Jeffrey A. Livermore, "Factors that impact implementing an Agile Software Development Methodology", pp. 82-85, IEEE 2007.
- [7] A. Ahmed, S. Ahmad, Dr. N. Ehsan, E. Mirza, S.Z. Sarwar, "Agile Software Development:Impact on Productivity and Quality" , pp. 287-290, IEEE 2010.
- [8] Ying Wang, Dayong Sang, Wujie Xie, "Analysis on Agile Software Development Methods from the View of Informationalization Supply Chain Management", 3rd International Symposium on Intelligent Information Technology Application Workshops", pp. 219-222, 2009.
- [9] Agile Methodology: An Overview - <https://zenkit.com/en/blog/agile-methodology-an-overview/>
- [10] An Introduction to Agile Methodology - <https://www.codeproject.com/Articles/704600/An-Introduction-to-Agile-Methodology>
- [11] Ashish Sangavi , Parth Sehgal , Avnish Kapur,Suyash Singh "Influence of Number of People in Agile Software Development" International Research Journal of Engineering and Technology (IRJET) Volume 4 Issue 1, Jan 2017
- [12] Gaurav Kumar, Pradeep Kumar Bhatia "Impact of Agile Methodology on Software Development

Process" International Journal of Computer Technology and Electronics Engineering Volume 2 Issue 4 August 2012

- [13]. Johnson, R., 2008. Six Principles of Effective Management. [Online]. Available: <http://ezinearticles.com/?Six-Principles-ofEffectiveTeam-Management&id=1803062>. [Accessed: 28th March 2011]
- [14]. Kemerer, C., 1989. An agenda for research in the managerial evaluation of computer-aided software engineering (CASE) tool impacts, Proceedings of the 22nd Annual Hawaii International Conference on System Sciences, Hawaii, pp. 219-28.
- [15]. Lencioni, P., 2002. The Five Dysfunctions of a Team. Jossey-Bass
- [16] Agile Architecture: Strategies for Scaling Agile Development - <http://agilemodeling.com/essays/agileArchitecture.htm>
- [17] An Agile approach to the Software Architecture - <https://www.agileconnection.com/article/agile-approach-software-architecture>
- [18] Towards an Agile Software Architecture - <https://www.infoq.com/articles/towards-agile-software-architecture/>
- [19] Scale Agile Framework © Scaled Agile, Inc.- <https://www.scaledagileframework.com/agile-teams/>
- [20] Figures of works with multiple teams and software architects :- <https://www.infoq.com/articles/towards-agile-software-architecture/>