

Agile Team Management: A Review

Anubhav Buragohain¹, Sasi Bhushan Saikia², Nomi Baruah³

Student, Computer Science Engineering, Dibrugarh University, Dibrugarh, India^{1,2}

Assistant Professor, Computer Science Engineering, Dibrugarh University, Dibrugarh, India³

Abstract: This paper involves in identifying how the team management is done in various Agile Methodologies. Management is a hugely important part of putting together and managing the software project teams. Agile approaches help teams respond to unpredictability through incremental, iterative work cadences and empirical feedback. Agile methodology provides opportunities to assess the direction of a project throughout the development lifecycle to a team. Agile Team Management has brought a revolution to the IT sector after its introduction by which companies are now able to maintain a healthy relationship with all its employees, thereby encouraging them to work efficiently with less stress and achieve desired goal within stipulated time. Many software projects are not successfully completed or cannot meet the user requirements due to lack of proper project management as organizations continue to diversify, the opportunities for workplace problems intensify. Often the underlying causes of these problems are the lack of open, owing communications or using the wrong organizational structure.

Keywords: Agile Methodology, Software Companies, Software Projects, Team Management.

I. INTRODUCTION

The Agile methodology seeks alternatives to traditional project management. Agile approaches help teams respond to unpredictability through incremental, iterative work cadences and empirical feedback. Agilists propose alternatives to waterfall, or traditional sequential development [1]. Agile software development methodology is a process for developing software. In English, Agile means ‘ability to move quickly and easily’ and responding swiftly to change – this is a key aspect of agile software development as well [2]. In agile methodology, the team is self-organizing. It sets its own schedule based on priorities from the product owner and the available capacity of the team [3].

Team Management refers to the various activities which binds a team together by bringing the team members closer to achieve the set targets [4]. Team management involves teamwork, communication and objective setting, capability to identify problems and resolve conflicts within a team [5]. Having a proper management team in a software industry helps to operate in different levels in different locations or divisions. There must be a proper management of the team in order to avoid failure [6]. The usage of the agile methodologies has been increased over the past few years for the development of software applications. A proper Team Management can lead to proper distributions of work among the team members that can help lead to a required goal [7].

II. AGILE METHODOLOGIES

i. Scrum

Scrum is a management and control process that cuts through complexity to focus on building software that meets business needs. Scrum itself is a simple framework

for effective team collaboration on complex software projects [8].

Within the Scrum Framework three roles are defined. The first is a Scrum Team that is a collection of individuals working together to deliver the requested and committed product increments [9]. The second is Scrum Master, who can be thought of as a coach for the team, helping team members use the Scrum process to perform at the highest level [10]. The last is the product owners who are the champions for their product. They are focused on understanding business and market requirements, then prioritizing the work to be done by the engineering team accordingly [11].

ii. Extreme Programming

Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements [12]. Extreme Programming implements a simple, yet effective environment enabling teams to become highly productive [13].

XP has several different roles, each with its own responsibilities and unique rights. XP attempts to improve communication between customers and developers. XP gives developers authority to make technical decisions. The customer drives the project. He defines the project and sets its goals [14].

iii. Scrumban

Scrumban is an Agile management methodology describing hybrids of Scrum and Kanban and was originally designed as a way to transition from Scrum to Kanban. Scrumban is a management framework that emerges when teams employ Scrum as their chosen way of working and use the Kanban Method as a lens through

which to view, understand and continuously improve how they work [15] [16].

The teamwork is organized in small iterations and monitored with the help of a visual board, similar to Scrum and kanban boards. To show each stages of work, teams working in the same space mostly use post-it notes or a large whiteboard. If the teams are decentralized, visual management software such as Assembla, Target process, Eylean Board, JIRA or Agilo for Trac are used [15]. Planning meetings are organized to determine the next user stories to be completed in the next iteration. The stories are then added to the board and the team completes them, the team working on few user stories at a time as practical. To keep them short, WIP limits are used, and a planning trigger is set for the team to identify when to plan next - when WIP falls below a predetermined level. There are no predefined roles in Scrumban; the team keeps the roles they already have [17].

iv. Kanban

Kanban (literally signboard or billboard in Japanese) is a scheduling system for lean manufacturing and just-in-time manufacturing (JIT) [18]. Kanban is an inventory-control system to control the supply chain. Taiichi Ohno, an industrial engineer at Toyota, developed kanban to improve manufacturing efficiency. Kanban is one method to achieve JIT [19].

In software development, Kanban provides a visual process-management system which helps in decision-making about what, when and how much to produce.

Target process helps Kanban-oriented teams improve their just-in-time delivery. It visualizes the transition of a task from creation to its delivery to the customer without overloading the team members.

v. Feature-driven development (FDD)

Feature-driven development (FDD) is an iterative and incremental software development process. It is one of the number of lightweight or Agile methods for developing software. FDD blends a number of industry-recognized best practices into a cohesive whole. These practices are all driven from a client-valued functionality (feature) perspective. Its main purpose is to deliver tangible, working software repeatedly in a timely manner [20].

There are six primary roles in FDD project: Project Manager, Chief Architect, Development Manager, Chief Programmer, Class Owner, and Domain Expert. An individual will take on one or more roles on a project. The concept of a class owner is where FDD differs from XP. XP has a practice called Collective Ownership i.e. any developer can update any artifact, including source code, as required. FDD takes a different approach in assigning classes to individual developers, so if a feature is required to be changed in several classes then the owners of those classes must work together as a feature team to implement it [21].

vi. Adaptive Software development (ASD)

Adaptive software development (ASD) is a software development process that grew out of rapid application development work by Jim Highsmith and Sam Bayer. It embodies the principle that continuous adaptation of the process to the work at hand is the normal state of affairs. It replaces the traditional waterfall cycle with a repeating series of speculate, collaborate, and learn cycles. This dynamic cycle provides for continuous learning and adaptation to the emergent state of the project. The characteristics of an ASD life cycle are that it is mission focused, feature based, iterative, timeboxed, risk driven, and change tolerant [22].

No specific team management is required.

vii. Agile Unified Process (AUP)

Agile Unified Process (AUP) is a simplified version of the Rational Unified Process (RUP) developed by Scott Ambler [23]. The AUP applies agile techniques including test-driven development (TDD), agile modelling (AM), agile change management, and database refactoring to improve productivity [24]. This methodology is a combination between the Rational Unified Process (RUP) and Agile Method (AM) [25].

AUP is scalable. In other words, it can be applied to small and large projects. The AUP lifecycle is serial in the large, whereas it is iterative in the small [26].

Team management in AUP is done by the Project Manager in which he manages the team, protects the team, build the relationship with stakeholder, interaction with stack holder, plan, manage and allocate resources, keep the team to be focused. The Developer in the other hand write, tests and builds the software.

viii. Dynamic Systems Development Method (DSDM)

Dynamic systems development method (DSDM) is an agile project delivery framework, primarily used as a software development method [27] [28]. DSDM is an iterative and incremental approach that embraces principles of Agile development, including continuous user/customer involvement [29].

In DSDM, the team management is done specifically by the directors, managers and coordinators of the project work. The Business sponsor will be providing funds and direction to the organization. The Business Visionary and Technical coordinator will work towards getting an excellent solution. The Project Manager will make sure that the funds allocated to the organization are used effectively to reach the desired outcome. Team Leader, Business Ambassador, Business analyst, Solution Developer and Solution Tester act as the engine room of the project to shape and build the solution, which 'are collectively responsible for day-to-day development and assuring fitness for business purpose' [30]. Solution testers will report to the Technical coordinator regarding the quality of the technical features involved in the project. Business Advisors, Workshop Facilitator and Atern Coach will be providing assistance and guidance to the project in a more generic way.

ix. Lean software development (LSD)

Lean software development (LSD) is a translation of lean manufacturing and lean IT principles and practices to the software development domain [31].

Lean development can be summarized by seven principles, very close in concept to lean manufacturing principles: [32]

- a. Eliminate waste
- b. Amplify learning
- c. Decide as late as possible
- d. Deliver as fast as possible
- e. Empower the team
- f. Build integrity in
- g. See the whole

Team management in lean software development requires the leader 'to use a matrix to find what the key processes are and determine who needs development, in what areas they need it and who is best placed to teach them those skills' [33]. The matrix is a very important tool which lets people to work together and develop the need for development to turn group into a team. It helps people to become closer and things become more balanced indicating that everyone will contribute to the work and will help each other in teaching something which the other individual may lack in.

III. CONCLUSION

This paper rigorously finds the team management approaches for software projects which uses Agile Methodology for software development. The continuous change in market demand and the zeal to conquer the market in the respective domain made the software organization to work with the agile methodologies.

REFERENCES

- [1] "Agile Methodology Understanding Agile Methodology," Agile Methodology RSS. [Online]. Available: <http://agilemethodology.org/>. [Accessed: 25-Dec-2016].
- [2] "ISTQB Exam Certification," ISTQB Exam Certification. [Online]. Available: <http://istqbexamcertification.com/what-is-agile-methodology-examples-when-to-use-it-advantages-and-disadvantages/>. [Accessed: 25-Dec-2016].
- [3] "Development managers vs. scrum masters," Atlassian. [Online]. Available: <https://www.atlassian.com/agile/effective-management-across-agile>. [Accessed: 26-Dec-2016].
- [4] MSG Management Study Guide, Team Management - Meaning and Concept. [Online]. Available: <http://www.managementstudyguide.com/team-management.htm>. [Accessed: 21-Dec-2016].
- [5] Team management, Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Team_management. [Accessed: 21-Dec-2016].
- [6] Develop your management team - infoentrepreneurs.org. [Online]. Available: m.infoentrepreneurs.org/en/guides/develop-your-management-team/. [Accessed: 21-Dec-2016].
- [7] [Online]. Available: www.mecs-press.org/ijeme/ijeme-v6-n5/IJEME-V6-N5-1.PDF. [Accessed: 21-Dec-2016].
- [8] Scrum.org | The home of Scrum, What is Scrum?, What is Scrum? | Scrum.org -The home of Scrum. [Online]. Available: <https://www.scrum.org/Resources/What-is-Scrum>. [Accessed: 21-Dec-2016].
- [9] Scrum Roles - The Scrum Team, Scrum Roles The Scrum Team International Scrum Institute. [Online]. Available: http://www.scruminstitute.org/Scrum_Roles_The_Scrum_Team.php. [Accessed: 21-Dec-2016].
- [10] @mikewcohn, Scrum Methodology and Project Management, Mountain Goat Software. [Online]. Available: <https://www.mountaingoatsoftware.com/agile/scrum>. [Accessed: 21-Dec-2016].
- [11] Scrum, Atlassian. [Online]. Available: <https://www.atlassian.com/agile/scrum>. [Accessed: 21-Dec-2016].
- [12] Extreme programming, Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Extreme_programming. [Accessed: 21-Dec-2016].
- [13] Extreme Programming: A Gentle Introduction., Extreme Programming: A Gentle Introduction. [Online]. Available: <http://www.extremeprogramming.org/>. [Accessed: 21-Dec-2016].
- [14] C., W. Cunningham, and T. A. Diaz, Extreme programming: pocket guide ;Beijing:O'Reilly, 2003.
- [15] Lean Software Engineering, Lean Software Engineering Atom. [Online]. Available:<http://leansoftwareengineering.com/kse/scrumban/>. [Accessed: 21-Dec-2016].
- [16] A.Reddy, The Scrumban revolution: getting the most out of Agile, Scrum, and lean Kanban. New York: Addison-Wesley, 2016.
- [17] <http://www.youtube.com/channel/U.C.I.B.P.D.G.W.S.X.Y.I.A.>, YouTube, YouTube, 2014. [Online]. Available: <https://www.youtube.com/watch?v=uaNHFz7Yw>. [Accessed: 21-Dec-2016].
- [18] The definition of kanban, Dictionary.com. [Online]. Available: <http://www.dictionary.com/browse/kanban>. [Accessed: 21-Dec-2016].
- [19] T. Ono, Toyota production system: beyond large-scale production. Cambridge, MA: Productivity Press, 1988.
- [20] Feature-driven development, Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Feature-driven_development. [Accessed: 21-Dec-2016].
- [21] Feature Driven Development (FDD) and Agile Modeling, Feature Driven Development (FDD) and Agile Modeling. [Online]. Available:<http://www.agilemodeling.com/essays/fdd.htm>. [Accessed: 21-Dec-2016].
- [22] Adaptive software development, Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Adaptive_software_development. [Accessed: 21-Dec-2016].
- [23] 28 Feb 2008 at 06:02, John K Waters tweet btn(), Agile lands role in games and business software. The Register. [Online]. Available: http://www.theregister.co.uk/2008/02/28/agile_crossing_chasm/. [Accessed: 21-Dec-2016].
- [24] Agile Unified Process, Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Agile_Unified_Process. [Accessed: 21-Dec-2016].
- [25] [Online]. Available: http://www.arpnjournal.com/jeas/research_papers/rp_2014/jeas_10-14_1248.pdf. [Accessed: 21-Dec-2016].
- [26] [Online]. Available: <http://www.slideshare.net/dcsunu/agile-open-unified-processes>. [Accessed: 21-Dec-2016].
- [27] K. Richards, Agile project management: running PRINCE2 projects with DSDM Atern. London: The Stationery Office, 2007.
- [28] G. Cantone and M. Marchesi, Agile processes in software engineering and extreme programming: 15th International Conference, XP 2014, Rome, Italy, May 26-30, 2014, proceedings. Cham: Springer, 2014.
- [29] Dynamic systems development method, Wikipedia. [Online]. Available:https://en.wikipedia.org/wiki/Dynamic_systems_development_method. [Accessed: 21-Dec-2016].
- [30] Roles and Responsibilities, Agile Business Consortium, 2016. [Online]. Avail-able: <https://www.agilebusiness.org/content/roles-and-responsibilities-0>. [Accessed: 21-Dec-2016].
- [31] Lean software development, Wikipedia. [Online]. Available: <https://en.wikipedia.org/wiki/Leansoftwaredevelopment>. [Accessed: 21-Dec-2016].
- [32] M. Poppendieck and T. D. Poppendieck, Lean software development: an agile toolkit. Boston, MA: Addison-Wesley, 2003.
- [33] W. Nijdam, A step-by-step guide to leadership and team lean development, PlanetLean. [Online]. Available: <http://planetlean.com/the-step-by-step-lean-development-of-teams-and-leaders>. [Accessed: 21-Dec-2016].