Rewarding High Performers in Scrum

Abstract—**In spite of gaining popularity for its productivity and quick turnaround time, organizations face various challenges in Agile based Scrum implementation such as unavailability of ProductOwner for distributed teams, unable to fit in certain personalities, missing some of important functionalities like reviewing, coaching, mentoring etc. To overcome these challenges, organizations tend to customize the Scrum framework based on their culture and work practices. This paper delves into some of these challenges observed in Scrum implementation for software development projects and offers a generic solution by differentiating high performers from rest others through a new role called "ScrumOwner" thereby improving overall Scrum productivity and building a strong technical leadership in the organization.**

Keywords-Agile software development; Distributed Agile; Scrum methodology; Scrum Ownership; Scrum Leadership; Issues in Scrum; High performers in Scrum

I. INTRODUCTION

With each passing day, Agile methodologies are gaining wide popularity and resulting into high adoption across industries. Scrum framework is one of the popular Agile methodologies adopted for software development projects. Since every organization has a unique culture and characteristics, it faces some interesting challenges while adopting and implementing the Scrum framework. Challenges become even more intense when geographically distributed members communicate within or across teams.

In most of the global software development organizations, requirements are primarily generated at one location and implemented at multiple locations. Such distributed software development environment poses challenges for inter-team communication. In addition to this, because of social nature [1] of the Scrum teams, some of the high performing individuals are not utilized effectively. This paper analyzes these challenges and proposes a solution that can be widely adopted for productive Scrum implementation across geographies. The paper revisits Scrum roles and advocates that the technical leadership within Scrum holds the key in overcoming these challenges. The solution presented here takes care of aspirations of high performing individuals and effectively accommodates them in Scrum framework by offering them increased responsibilities.

Scrum framework consists of Scrum teams and their associated roles, time boxes, artifacts and rules. Scrum teams are designed to allow flexibility and optimum productivity. Each Scrum team has three roles: the ScrumMaster, the ProductOwner and the team. With regard to productivity and performance, Scrum emphasizes team effort rather than the individual contribution [9].

A scrum team typically consists of five to ten people responsible for delivering the features described in high level requirement documents. The team should be self-governing, self-organizing, and cross-functional. A ScrumMaster is responsible for ensuring that the team adheres to Scrum values, thereby facilitating meetings, coaching and mentoring them about Scrum practices. The ProductOwner is the most demanding role responsible for overall product success. ProductOwner has a long term vision about a product and understands its business value.

Scrum employs various time boxes for executing and monitoring the work progress via Release Planning Meeting, Sprint, Iteration, Sprint Planning Meeting, Daily Scrum and Sprint Review and Retrospective Meeting.

II. RELATED WORK

Organizations implementing Scrum framework reported various issues and customized the framework as per their culture and work practices. Mike and Marcus [2] observed wastage of time and resources due to unavailability of a single ProductOwner for all Scrum teams. In order to overcome this challenge, they experimented with the ProductOwner role by abolishing the central role, creating multiple local ProductOwners and allowing team to take care of ownership of respective features. Eric and Rajani [6] reported issues about dependency management and about reviewing functional designs and technical implementation in a large Agile environment. These issues were handled by creating virtual architecture team of cross-functional and cross-scrum team members and conducting design review meetings. Elizabeth and Robert [1] observed issues when an individual disowns self-identity and continuously engages in a team environment or "being socially active".

III. DESIGN OF EXPERIMENT

A research study was conducted in order to understand how Scrum teams perform under various operating conditions and to ascertain if there were significant differences in their performance and self-governing nature. A software development project having activities organized in various scrum teams across different geographies was chosen and its activities were closely monitored.

Each scrum team consists of a set of dedicated developers, testers, a ScrumMaster and a ProductOwner. All scrum teams operate simultaneously within the same iteration duration of about three to four weeks. Dependency management and cross team coordination were handled through bi-weekly Scrum-of-Scrum meetings. One set of scrum teams were co-located along with ProductOwners; the other set of scrum teams operated from remote locations, usually crossing time zones and having less access to ProductOwners. These teams are called as local teams and remote teams respectively in the further discussion. Local teams were established earlier in the product life cycle than remote teams and had higher product knowledge than their remote counterparts.

Burndown charts of these two sets of teams are monitored, aggregated and compared for the development period of the project to find out the differentiating factors. Fig. 1 shows the burndown charts for local and remote teams. An ideal burndown chart is also drawn for the comparative purpose. Following observations are drawn from the comparative study.

- There is a significant difference in the way local and remote teams work in a given iteration as demonstrated by the shapes of burndown charts.
- The burndown chart for local teams closely resembles to the expected burndown where as for the remote team it is not as per the expectations.
- Local teams seem to get started with the iteration activities right on the first day of the sprint where as remote teams spend considerable time in assimilating requirements and planning activities.
- Local teams define the content early in the iteration and deliver it as per the expected pace.
- Remote teams spend initial few days for identifying the content and deliver it in later phase with faster pace than expectations.

IV. Challenges in Scrum

Further study is conducted to investigate the differences in working style of local and remote teams and it leads to following interesting facts.

A. Self-governing nature

The comparative chart in the current experiment clearly demonstrates that remote teams lack the self-governing nature as expected by Scrum philosophy. Scrum process emphasizes the collective ownership rather than the individual ownership. However, sometimes collective ownership ends up with no ownership. ScrumMaster and the team should behave in an empowered manner whenever there are any decisions to make. Relying on majority rules results in safe decisions and a mediocre product but sometimes the best answer is to let an individual win over the majority [3]. Team members can have diverse set of opinions about a point and it is tough for ScrumMaster to arrive at a decision. Teams often spend considerable amount of time in sprint planning in order to accommodate everyone's point of view which appears as if the collective ownership doesn't work or the team lacks the self-governing nature.

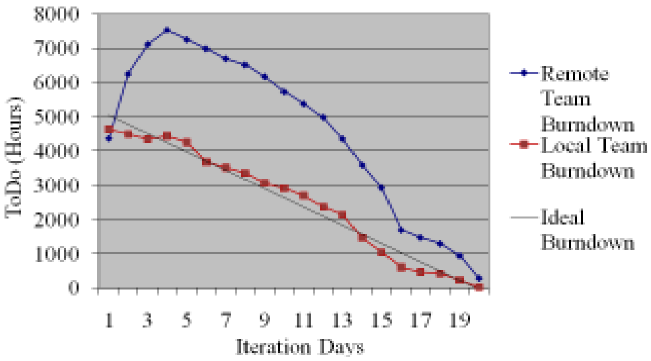

Figure 1. Comparative Burndown of Local and Remote teams

B. Non availability of ProductOwner

In the current experiment, local teams had the advantage of being close to ProductOwner where as remote teams spend considerable amount of time, usually stretching office hours in chasing ProductOwner for release planning meetings and sprint planning meetings. This results into a burnout feeling among the team members and eventually a delay in burndown activities.

Any delay in acquiring requirements leads to wastage of time and resources during initial few days of the iteration. Some proactive teams take decisions based on their knowledge and experience and move ahead in order not to waste time. Eventually when ProductOwner comes later with the right direction, the earlier work either gets scrapped or needs correction [2]. The ProductOwner role turns out to be a bottleneck for the success of the release.

C. Coaching and Estimating

The comparative chart in the current experiment shows that local teams accurately forecast and estimate the efforts involved in first couple of days of the iteration where as remote teams take some extra amount of time.

The success of Scrum lies entirely in the self-governing and self-organizing nature of team members. This is quite a significant assumption in formulating Scrum methodology. However, in practice, team members have varying levels of expertise in product domain and technology. Some novice members look for expert advice in these areas and also in development and testing processes. ScrumMaster does not necessarily offer coaching and mentoring in areas that the team is expecting; ScrumMaster's coaching is limited to just Agile practices and daily rituals.

All Scrum members are expected to decompose the sprint backlog into smaller tasks that can be completed in less than a day and also provide the estimates. Some of the new entrants in the team may not have acquired such skills. Wrong or untrue estimates lead to unexpected results, surprises or even complete failure at the end of the sprint. Unless equipped with adequate technical expertise, ScrumMaster does not necessarily offer expert advice in accurate estimation and planning.

D. Social nature of Scrum Teams

In Scrum framework, productivity and performance are measured with respect to a team's collective efforts rather than individual contribution. A team is usually consists of varying degrees of expertise among members, some high performing and some low performing ones. Cohesive Scrum teams often close the gap by averaging individual expertise and assigning limited responsibilities to all team members. Such pessimistic approach often lead to inefficient and underperforming teams [3] through following few reasons: unable to transition unique individual style into Agile idealism; high performing individuals not utilized to the best of their abilities; stress or exhausted feeling after being socially active for the whole iteration [1].

E. Leadership vs. Management

Marcus Buckingham [7], in his book "The One Thing You should Know", differentiates two distinct yet complementary organizational traits, leadership and management. Leadership is the ability to create solid vision for the team he/she is leading where as management is the ability to match people's task with their skills in order to achieve the defined goals [7].

Scrum framework personifies the leadership in the form of ProductOwner for the entire product and Scrum teams are left to chalk out their own visions by creating visionlet, a light weight functional specification document. The ScrumMaster role by its definition does not offer leadership but rather closely resembles to a managerial role of removing impediments, matching people skills, monitoring and tracking planned activities where as a ProductOwner may not be able to set vision for all Scrum teams simultaneously. Higher management or Scrum-of-Scrum has to step-in to induce leadership at Scrum level.

F. Career Aspirations

Most of the roles in software organization are divided into two categories, technical and managerial. Although recently, some organizations are trying to close the gap between the two skills, every role in the organization exhibits one dominant skill over the other, either technical or managerial. Laying out systematic career path for either of the skills right from the bottom level to the top level is vital for any organization. Scrum literature talks about technical leadership roles within the organization like Architect, Product Manager or Business Analyst and explains how these roles get molded in the Scrum framework; however the framework is equally silent over the path or the process that led an individual to take up these higher roles. Thus, members aspiring career in one of the skills might not visualize career roadmap and in turn get de-motivated.

V. INTORODUCING SCRUMOWNER ROLE

The concept of a local ProductOwner as introduced by Mike and Marcus [2] is extended here in the form of a “ScrumOwner”. Scrum-level ProductOwner is a subject matter expert at team level and is responsible for aligning the team in right direction. In addition to the team level subject matter export role, ScrumOwner offers technical leadership, coaching and mentoring abilities in the technology and the domain of the product. Following are few of the attributes and responsibilities of the ScrumOwner role.

- The ScrumOwner role is played by a person who is fully engaged in the development or testing activities and who knows the Scrum deliverables from end to end. This person is supposed to own the sprint content from technical or functional point of view; ScrumMaster still continues to own the content from people and delivery point of view. This separates technical and nontechnical responsibilities between the two roles.
- ScrumOwner interacts with ProductOwner continuously during a sprint for breaking down the product vision into smaller feature and prioritizing them for subsequent sprints. Early planning and preparation for next sprint eliminate wastage of time and resources and also wrap sprint planning meeting quickly in time.
- ScrumOwner encourages and guides team members in publishing visionlets about the feature. However, if no single person is experienced enough or if no one takes the initiative, ScrumOwner demonstrates the leading capability by publishing and circulating visionlets to all the stakeholders.
- In case of limited or unavailability of ProductOwner, ScrumOwner acts as a local ProductOwner or Architect responsible for identifying and prioritizing the contents, defining acceptance criteria and also verifying the sprint deliverables.
- ScrumOwner acts as an intermediate layer between ProductOwner and the team by clarifying and answering trivial queries with respect to the requirements. It helps ProductOwner not to indulge in less important activities and concentrate on larger product vision.
- ScrumOwners from different Scrum teams interact with each other and address technical issues concerning all Scrum teams, for example, laying out best practices around architecture, design, development, testing, designing and code reviewing across the entire product.
- ScrumOwner offers technical guidance to resolve any blocking issues the team is facing during the iteration.
- ScrumOwner provides coaching and mentoring to novice members in the technology and the domain, assists them in breaking the requirement into smaller tasks and guides them in correctly estimating their tasks. ScrumOwner also imparts best coding and designing practices within the Scrum team.
- ScrumOwner acts as a default candidate performing design and code review for performance, scalability and maintainability of the feature.
- ScrumOwner can be an indispensable resource for organizations undergoing staffing changes in order to rejuvenate the team in the changed scenario.
- ScrumOwner assists ScrumMaster in resolving conflicts and in taking important decisions.

VI. Benefits of the Improved Team Structure

Introduction of the ScrumOwner role can have huge impact in effective talent utilization and also in building strong technical leadership within the organization.

The proposed approach nicely differentiates leading vs. managing tasks and induces ownership at Scrum level. ScrumOwner is a single point of contact in dealing with technical nitty-gritty of the feature where as ScrumMaster is a single point of contact for staffing and delivery related issues.

The knowledge gap between ProductOwner and the team is greatly reduced by having intermediate role of ScrumOwner. It boosts ProductOwner's confidence with the team and improves communication among all stakeholders. Since ScrumOwner is continuously engaged with ProductOwner throughout the iteration, the team would be in a better position to plan and estimate the activities in first few days of the iteration.

The enhanced Scrum structure channelizes the career aspirations of individuals and facilitates them to make conscious decisions in accepting roles as per their interests and capabilities. Consistent performance in the ScrumMaster role leads towards higher roles in management track and consistent performance in the ScrumOwner role leads towards higher roles in technical track such as Product Manager, Architect and Business Analyst. This induces greater flexibility in career choices and employees can be freely rotated between these two roles based on their choices and the organizational needs while still operating within the Scrum environment.

High performing individuals can take up the ScrumOwner role and offer coaching and mentoring to team members in technical areas which otherwise is not possible in normal Scrum framework. Availability of expert advice helps team members in planning and estimating their tasks and eventually minimizes surprises at the sprint end.

The introduction of the ScrumOwner role takes care of variance of expertise and addresses the social aspect of team working. Technical team members with great velocity at individual level can be rewarded with a ScrumOwner role while still being part of the team. Differentiating from rest of the team and giving additional responsibilities motivates them for higher performance [10] and at the same time, inspires other team members for higher performance.

The ScrumOwner role can be rotated among team members in order to build the new breed of technical leaders. Similarly, ScrumOwner can also be moved to other teams to take charge of new assignment and to expand technical boundaries. Such a rotation policy unleashes new opportunities, induces "Knowledge Agility" at individual level and promotes healthy competition in the organization.

Scrum teams with a ScrumOwner would be in a better position to absorb any organizational changes than those having just a ScrumMaster. ScrumOwners are the best possible resources in bringing newcomers up to a productive level, rejuvenating the team in the changed organizational structure and thereby accelerating the Scrum performance.

Geographically distributed Scrum teams with ScrumOwners can quickly get start with the iteration activities without waiting for ProductOwner's availability and do not waste time during initial period of the iteration.

VII. Applicability of ScrumOwner

Introduction of ScrumOwner should be a conscious decision and at most care should be taken not to create multiple power centers within Scrum. The ScrumOwner role should be introduced without having explicit hierarchy within the team and without disturbing the team synergy. Introduction of the ScrumOwner role may not be suitable when the team members are equally mature and self-organized. A person with adequate technical and managerial capabilities can simultaneously play ScrumMaster and ScrumOwner roles without explicitly introducing the new role.

VIII. Conclusions

The current experiment demonstrates some of the pitfalls of Scrum based software development and proposes a solution for addressing most of the commonly observed challenges. Separation of product and people oriented tasks into different roles helps in motivating the high performers as well as building strong technical leadership within the organization. Introduction of the "ScrumOwner" role unleashes new career opportunities for technical experts, induces "Knowledge Agility" at personal level and imparts healthy competition in the organization. The insights presented here can be adapted by implementing ScrumOwner in real practice and observing the performance trend in the enhanced Scrum team structure.

References

[1] Elizabeth Whitworth and Robert Biddle, "The social nature of agile teams", In Proceedings of the Agile Software Development Conference, Washington D.C., doi: 10.1109/AGILE.2007.60.

[2] Mike Lowery and Marcus Evans, "Scaling Product Ownership", Agile 2007, doi: 10.1109/AGILE.2007.51.

[3] Mike Lowery and Marcus Evans, "Great Scrums Need Great Product Owners: Unbounded Collaboration and Collective Product Ownership", Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, doi: 10.1109/HICSS.2008.186.

[4] Juyun Cho, "Issues and Challenges of Agile Software Development with Scrum", Issues in Information Systems VOL IX, No. 2, 2008 pp. 188-195.

[5] J. R. Barker, "Tightening the Iron Cage: Concertive control in self-managing teams", Administrative Science Quarterly, 38(3), 1993, pp. 408-437.

[6] Eric Babinet and Rajani Ramanathan, "Dependency Management in a Large Agile Environment", Agile 2008, doi: 10.1109/Agile.2008.58.

[7] Marcus Buckingham, "The One Thing You Should Know", Free Press.

[8] Angela Martin, James Noble and Robert Biddle, "Experience on the Human Side of Agile", doi: 10.1007/978-3-540-68255-4. Springer Berlin Heidelberg.

[9] Ken Schwaber, "Scrum Guide".

[10] Nitin Nohria, Boris Groysberg and Linda-Eling Lee, "Employee Motivation: A Powerful New Model", Hardward Business Review, July-August 2008, doi: 10.1225/R0807G