International Journal of Business Quantitative Economics and Applied Management Research

Volume-7, Issue-2, 2022 ISSN No: 2349-5677

TRANSITION OF PROJECT MANAGEMENT FUNCTIONS TO AGILE SOFTWARE DEVELOPMENT MANAGEMENT

Santosh S Deshmukh
PMP, SCM, MSBA,Senior Member, IEEE

Abstract

The role of Project Managers in agile software development (ASD) has been a topic of debate. The empirical process and Agile manifesto decline the place for project managers in the Agile framework. However, Project Managers are mostly considered a "command and control" style of managing projects, whereas ASD encourages self-management and servant leadership which is not so common in traditional Project Management. There is no straight answer for the existence of a Project Manager in ASD but it's a choice with many impacting parameters. On the other hand, enforcing the Project Management role in ASD can increase the chances of ambiguities, and conflicts or can raise real concern. The research also shows that pre-defined roles in ASD are often assigned with project management roles but without project managers, like Product owner, scrum master, etc. This explains Project Management duties are indispensable, but the Project Manager's role is not. Some research shows Project Managers can still play a critical role in Agile environments, albeit with adjustments to their responsibilities and mindset. This paper explains how the conventional Project Management functions are mapped or managed when any organization transitions to Agile.

Index Terms— Project Management, Agile Software Development, Kanban, Scrum, Scaled Agile, Project Manager, Risk

I. INTRODUCTION

Project Management, as defined by the Project Management Institute (PMI), is a disciplined approach to initiating, planning, executing, controlling, and closing the work of a team to achieve specific goals and meet specific success criteria at the specified time. This traditional form of project management emphasizes the importance of careful planning, comprehensive documentation, and adherence to predefined project phases, including initiation, planning, execution, monitoring and controlling, and closure. The PMI approach is built on the premise that detailed planning and control can lead to the successful completion of project objectives, with a significant focus on aligning project goals with business objectives, managing resources, risk management, and stakeholder communication.

On the other hand, Agile software development, as articulated in the Agile Manifesto, offers a contrasting approach to project management and software development. Agile prioritizes individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan[4]. This methodology is characterized by its flexibility, iterative and incremental development, and focus on delivering value to the customer early and continuously throughout the project. Agile methodologies, such as Scrum, Kanban, and Extreme Programming (XP), encourage collaborative efforts among self-organizing and cross-functional teams, regular reflection on how to become more effective, and adaptation of behavior accordingly. Agile aims to break down projects into small, manageable pieces, allowing teams to adjust to changing requirements and deliver high-quality products more quickly and efficiently.

The juxtaposition of Project Management per PMI and Agile software development methodologies reflects a spectrum of approaches to managing projects, with PMI embodying a structured, process-driven approach and Agile advocating for flexibility, customer centricity, and adaptability. While PMI's methodology provides a comprehensive framework suitable for a wide range of industries and projects, requiring detailed planning and control, the Agile Manifesto's principles cater to the dynamic nature of software development projects, where requirements can change frequently, and rapid delivery of functional product increments is crucial. Both methodologies have their strengths and are best selected based on the project's specific needs, organizational culture, and the environment in which the project operates, demonstrating the diversity of approaches available for managing projects in various contexts.

The goal of this paper is not to discuss the pros and cons or pitfalls of either of these models but to understand the viewpoint of the Project Manager Role functions and how they are positioned in ASD considering the success of project delivery.

II. ADJUSTMENT OF KEY FUNCTIONS

Consider a small theoretical use case, a software organization wants to shift its project delivery office from a contemporary waterfall model to an agile model. The company has great belief in the contemporary project delivery model and is very successful in implementation. There are a bunch of parameters that affected this decision that company executives thought of, but the most alarming concern is to transition the project manager's responsibilities to an agile framework. Below are the key Project Management functions and how are those transformed into Agile methodology.

The conventional Project Manager's role is to manage the following project constraints	**How are these constraints transformed in Agile Software Development?**
Scope Management	In Agile, the scope is managed through an iterative, flexible process that accommodates change and focuses on delivering value to the customer in short cycles or sprints. The product owner is identified as a responsible person for defining the requirements in terms of stories (a chunk of

	requirements divided into small doable features). Review and prioritization is managed in the backlog, gold plating avoidance, tracking to completion, scope, creep, requirement stability is Product Owner functions.
Schedule/Time Management	Agile projects manage time through fixed-length iterations or sprints, continuous planning, and regular adjustments based on team velocity and stakeholder feedback. In Agile Scrum, every assigned story is the responsibility of the developer to finish in the committed time, normally it's a 2-week sprint. Scrum Master helps agile stakeholders remove the impediments, and blockers to ensure timely completion of stories. In case of non-completion, the spillover is added to the backlog and groomed for prioritization. Work is time-boxed, and it is everyone's responsibility to DONE planned work in Sprint.
Cost/Budget Management	The focus in Agile Scrum is on delivering value through iterative and incremental development cycles, known as sprints. The measurement of cost, therefore, aligns with these cycles and the Agile principles of flexibility, adaptability, and continuous improvement. The cost of the project is measured in terms of the efforts required for each story point. The estimated cost is nothing but the accumulated Story Point of the project. However, this gives only the labor cost to complete the said work. The other costs like travel, operations, and purchases is estimated separately.
Quality Management	In Agile projects, quality management is not a separate activity, but a collective responsibility embedded in every aspect of the project lifecycle. It is the responsibility of the whole team to manage the quality by testing the developed piece of functionality, continuous feedback, Definition of Done, and accurately outlining the acceptance criteria.

Table 1- Mapping of conventional project management functions to agile methodology

As we see here the four main constraints (also called as PMI project management knowledge areas), the responsibilities are segregated between the Product owner, Scrum master, and other Agile team members.

III. OTHER KNOWLEDGE AREA FUNCTIONS

In Agile methodology, the management of integration, procurement, stakeholders, risk, communication, and resources is woven into its iterative, collaborative, and flexible approach. Unlike traditional project management frameworks, Agile emphasizes adaptability, team collaboration, and customer feedback, impacting how these aspects are handled.

A. Integration Management

Integration in Agile is managed through continuous planning, integration, and evaluation processes. The iterative nature of Agile—working in sprints—allows for frequent reassessment and integration of project components. Daily stand-ups, sprint reviews, and retrospectives facilitate the seamless integration of new features and ensure alignment with project objectives. Agile teams use tools like Continuous Integration (CI) to automate the merging and testing of code changes, ensuring a cohesive product evolution.

B. Procurement Management

Procurement in Agile projects is often more flexible and adaptive, focusing on establishing partnerships with vendors that can support an iterative, change-welcoming process. Contracts may be structured to allow for changes in scope, emphasizing collaboration and value delivery over fixed deliverables. Agile procurement seeks vendors and tools that enhance the team's agility and responsiveness to change, rather than traditional fixed-scope, fixed-price agreements. When a project needs procurement either this can be handled by the project manager, or the agile team assigns the story to the procurement team (Scaled Agile SAFe).

C. Stakeholder Management

Stakeholder management in Agile is characterized by close collaboration and continuous communication. The Agile Manifesto values customer collaboration over contract negotiation, encouraging regular feedback and adjustments based on stakeholder input. Product owners play a key role in bridging stakeholders and development teams, ensuring that the project continuously aligns with business needs and stakeholder expectations. Regular demos and sprint reviews involve stakeholders, fostering transparency and trust.

D. Risk Management

Risk management in Agile projects is an ongoing activity integrated into the daily workflow. Agile methodologies embrace change, allowing teams to quickly respond to risks as they emerge. Regular sprint retrospectives and planning sessions are opportunities to identify, assess, and address risks. The practice of delivering work in small, manageable increments also helps in mitigating risks by making issues easier to identify and rectify early on. Scrum teams manage the risk and dependencies at story levels and PI (program increment) levels.

E. Communication Management

Communication in Agile is informal, frequent, and emphasizes direct interactions over comprehensive documentation. Daily stand-ups sprint planning, and retrospectives ensure that information flows freely within the team and with stakeholders. Information radiators, such as Kanban boards and burn down charts, provide visual communication of progress and impediments. Agile tools and practices aim to foster a culture of open communication, enabling quick decision-making and adjustments.

F. Resource Management

Resource management in Agile focuses on team dynamics and leveraging team strengths. Agile teams are typically cross-functional, self-organizing units that manage their own workload and make collaborative decisions on task allocation based on the team's capacity and skill set. Agile practices, such as pair programming and swarming, are used to optimize resource utilization, knowledge sharing, and skill development. The emphasis is on creating a sustainable work pace to maximize long-term productivity and team morale.

In ASD, many times Scrum Master Role is misunderstood as the Project Manager role. The Scrum Master role is a servant leader role that unblocks the impediments, does a lot of documents, and ensures the team meets the definition of done. The Scrum Master and Product Owner share most of the Project management responsibilities with the difference that these responsibilities come in chunks and an iterative fashion, so they are manageable. However, there are some similarities between the Scrum Master and Project Manager roles, both should have strong organizational and communication skills. Transitioning to either of the roles needs a strong change in mindset, as both roles function differently.

IV. ADJUSTMENT TO THE ROLE

As our intention of this research is to transition from an empirical Project Management role to an Agile servant leadership style, below are a few adjustments.

A. **Embracing Agile Values:** Project Managers need to embrace Agile values and principles, fostering a culture of collaboration, adaptability, and continuous improvement.

B. **Shifting from Directing to Facilitating:** The role shifts from one of command and control to facilitating team success, empowering team members, and encouraging ownership and decision-making within the team.

C. **Continuous Learning and Adaptation:** Agile Project Managers should be committed to continuous learning and adaptation, reflecting on processes and practices to identify areas for improvement.

D. **Develop Coaching Style:** Agile Project Managers should mold themselves more into a coaching role and encourage teams to be self-efficient and organized.

However, the above functions can change a little bit when the Project Manager is adjusted to the Product Owner role in ASD. The product owner defines the vision of the product; hence he/she can drive the requirements, draw the product roadmap, prioritize the backlog, etc., which normally a Project Manager ensures for requirement stability in the traditional project management style.

V. CONCLUSION

In conclusion, while the role of Project Managers in Agile Software Development evolves, they are not rendered obsolete. Instead, their role can be pivotal in ensuring that Agile teams operate effectively within the larger organizational context, bridging gaps, facilitating processes, and contributing to strategic planning and risk management. Their adaptability, leadership, and understanding of Agile methodologies can enhance the team's ability to deliver value efficiently and effectively. One of the important attributes a Project Manager should learn transitioning to Agile methodology is servant leadership. The organizations that transition from traditional project management style to Agile Software Development need to assess and align several parameters to answer questions like, is this transition necessary? Can this transition be partial to only a selective line of business or products? Is the organizational mindset ready for this transition? Is the organization ready for a big shift and ready to sustain its impact?

APPENDIX

ASD- Agile Software Development

PMI- Project Management Institute

SM – Scrum Master, a facilitator for an Agile development team.

PO - Product Owner, responsible for prioritizing and overseeing the development team's tasks.

SAFe- Scaled Agile Frame work, is a set of organizational and workflow patterns for implementing agile practices at an enterprise scale. The framework is a body of knowledge that includes structured guidance on roles and responsibilities, how to plan and manage the work, and values to uphold.[7]

DoD – Definition of Done, is an agreed-upon set of items that must be completed before a project or user story can be considered complete.[5]

XP - Extreme Programming. Extreme Programming (XP) is an agile software development framework that aims to produce higher-quality software, and higher quality of life for the team.[8]

REFERENCES

1. Alzoubi, Gill (2014). Agile global software development communication challenges: A systematic review
2. Cho, J.J. (2010). An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum. All Graduate Theses and Dissertations, 599.
3. Key Scrum Benefits in Project Management, https://www.workamajig.com/blog/scrummethodology-guide/scrum-benefits
4. Unhelkar, B. (2016). The Art of Agile Practice. CRC Press. Retrieved from: https://www.perlego.com/book/1471504/the-art-of-agile-practice-a-composite-approachfor-projects-and-organizations-pdf
5. Maier, P. (2017).Towards a Secure SCRUM Process for AgileWeb Application Development 73(1-8)
6. Kashyap S. (2018). Demystifying the 5 Phases of Project Management. Retrieved form https://www.proofhub.com/articles/project-management-life-cycle-5-phases.
7. What is SAFe ? Retrieved form https://scaledagile.com/what-is-safe/
8. eXtreme Programming (XP). Retrieved from https://kanbanzone.com/resources/agile/extreme-programming-xp/