5-2018

Project Management Assurance in Agile Projects: Research in Progress

Dawn Owens
University of Texas at Dallas, dawn.owens@utdallas.edu

Jeffrey W. Merhout
Miami University, emerhoujw@miamioh.edu

Deepak Khazanchi
University of Nebraska at Omaha, khazanchi@unomaha.edu

Project Management Assurance in Agile Projects: Research in Progress

Dawn Owens
University of Texas at Dallas
dawn.owens@utdallas.edu

Jeffrey W. Merhout
Miami University
emerhoujw@miamioh.edu

Deepak Khazanchi
University of Nebraska at Omaha
khazanchi@unomaha.edu

Project Management Assurance in Agile Projects:

Research in Progress

ABSTRACT

It is well known that many information technology (IT) systems development projects fail on one or more of the key criteria of cost, time and functionality. To manage the inherent complexity in IT projects and to provide methods for assessing project performance, we propose implementing an assurance process for IT projects - projects specifically using Agile/Scrum principles and practices. Building on prior work by Khazanchi and Owens (2010), the assurance process accounts for the uniqueness of the project while assessing potential risk factors at each stage of the project life cycle. We use the case of a midsized Midwestern U.S. university to gather feedback about how the project management assurance framework might work in the Agile project management environment.

Keywords

IT Project management, Project Management Assurance (PMA), Agile, Scrum, IT governance

MERINTRODUCTION

Information Technology (IT) projects are infamously difficult to manage and many of them result in failure. According to the Standish Group[1] (2015), two out of three IT projects fail and do not adequately deliver planned outcomes, resulting in significant consequences for organizations. These projects often lead to cost overruns, schedule overruns, and unmet requirements, while others are complete failures (ibid). In fact, it has been estimated that each year project failures cost the US economy in the range of $50 to $150 billion (Hardy-Vallee, 2012). As a result, failed projects can be detrimental to the bottom line by impacting a company's competitive advantage and customer perception.

A paradigm shift in software development and project management is the reliance on adaptive software development methodologies to manage uncertainty in projects. Adaptive software methodologies such as Agile methods, provide for continuous adaptation of the process to the changing nature of the project (Highsmith, 2000). Adaptive methodologies are designed around uncertainty due to increased variability of actual quality, time, and cost performance, and this makes managing risk crucial to the success of a project (Cagliano et al, 2015). Despite new age project management methodologies such as Scrum, project failure rates remain high. Failure to adequately address risk contributes to these failure rates causing budget overruns, schedule delays, and missed performance targets (Charette, 2005; Glass, 2006; Carbone & Tippet, 2004).

We adapt the notion of Project Management Assurance (PMA) proposed by Khazanchi and Owens (2010, 2017) to demonstrate how risks in Agile projects can be managed. This application contributes to the development of an effective

[1] https://www.infoq.com/articles/standish-chaos-2015.

methodology for mitigating risks in agile projects and therefore, increasing project success. The goal of this research is to offer PMA techniques to assure that Agile software development principles have been applied. In this regard, the aims of the paper are twofold:

- Adapt the Khazanchi and Owens PMA model to meet the needs of an Agile development methodology by presenting potential project risks and how they can be managed using Agile practices.
- Set the state for future research by integrating PMA and Agile practices and validating the process in practice.

In the next sections, we first discuss the background frameworks for PMA and Agile. We then show how project management assurance practices can be mapped to Agile practices, followed by a discussion of contributions and next steps.

AGILE FRAMEWORK

Agile software development methodologies are defined under the Agile Manifesto and arose out of a need to provide alternative methods to heavyweight, traditional waterfall software development methodologies (Agile Manifesto, n.d.). Agile methodologies value change and are designed to respond to changes (Agile Manifesto, n.d.). They acknowledge the fact that requirements change and being able to adapt to change is critical for success in software development projects. Scrum is an Agile framework for completing complex projects (Scrum Alliance, n.d.). It is an implementation of the values and principles of the Agile Manifesto. Scrum provides concrete steps for project teams and is designed to inspect and adapt feedback loops to cope with complexity and risk (Scrum Methodology, n.d.). Scrum uses ceremonial roles and meetings and operates in fixed durations or loops called **Sprints**. The Scrum framework consists of roles, events, artifacts, and rules. Scrum has three roles: **Product Owner** is responsible for communicating the vision and business value; **Scrum Master** is a facilitator between the product owner and team and guides the team through Scrum while helping to resolve issues; a **Team** is generally seven dedicated individuals responsible for self-managing and organizing to complete work. Important scrum artifacts are the **Product Backlog** and the **Spring Backlog.** During the planning phase of a project, a Product Backlog is created which contains a set of requirements, written as user **Stories** with business values. The Sprint planning team commits to complete a certain number of tasks that are managed in a Sprint Backlog. At the end of each Sprint the team will deliver a product and a Sprint **Retrospective** will occur to allow team members to reflect on the sprint and plan for improvements in the next sprint. Another unique characteristic of Scrum is the concept of a daily **Standup** meeting where each team member reviews progress toward the sprint goal. Scrum uses feedback looks to inspect the product and adapt to cope with complexity and risk. However, managing risk continues to be a problem. Therefore, we recommend utilizing an assurance process to assure that ceremonial roles and events have been followed. Next, we will summarize concepts from the project management assurance framework derived from the work of Owens and Khazanchi (2017).

PROJECT MANAGEMENT ASSURANCE

Project Management Assurance (PMA) evolved from software quality assurance models with an emphasis on risk management. We define project management assurance (PMA) as ***a set of assurance activities integrated into the information technology (IT) project management lifecycle*** (Khazanchi & Owens, 2010; Owens & Khazanchi, 2017). The objective of PMA is to assure successful project outcomes by reducing risk, assessing internal controls, and improving quality while confirming to the stated schedule and budget constraints. While software quality assurance focuses primarily on assuring quality software, project management assurance focuses on internal controls while assuring adherence to software standards and procedures (Owens & Khazanchi, 2009). PMA is not a single event, but a continuous process aimed at providing assurance throughout the project lifecycle. The unique characteristics of the PMA processes are outlined below.

- **Integration:** Assurance Activities are integrated with existing project management processes
- **Controls:** Assess and manage project performance through the reinforcement of established controls. We define controls as a system of procedures, mechanisms, or policies that could proactively prevent and detect IT project failures.
- **Continual Review:** A continual review of the inputs and outputs of each project phase.
- **Assessment:** Performed by an internal group, independent of the project team

PMA is unique in that it focuses on processes and deliverables rather than software while emphasizing controls and risks. Software quality assurance is aimed at reducing defects; project management assurance is aimed at identifying defects in the process (op. cit.). The process is performed by an external group without the fear of repercussions.

PMA is not the same as IT governance. IT governance is a subset of corporate governance that focuses on aligning investments in IT that support business strategy as well as controlling the IT function and assets in a way that meets all external compliance requirements, such as laws and regulations (e.g., securing personal identifying information) and also complies with internal policies and procedures that manage IT. The systems development methodology is a key part of an organization's IT governance that ensures that only needed IT projects are administered and that these projects are successful on a variety of criteria, such as budget, time and requirements. Thus, PMA is a component of governance that helps provide confidence in the development methodology. In the next section we summarize the findings of our work with a Midwestern university's IT Services personnel where we determined how Agile/Scrum practices address the risks inherent in IT projects.

PROJECT MANAGEMENT ASSURANCE: MAPPED TO AGILE PRACTICES

To address risks throughout the project, PMA divides risks into categories based on each of the project lifecycle phases and by project characteristics. The following tables illustrate risk factors as they relate to project characteristics and risk factors found during project initiation. We then show how they are applicable to Agile Practices.

Risk Factors related to Project Characteristics	**Applicable Agile Practices**
Project size – the larger the expense, staffing levels, elapsed time, and number of departments affected by the project, the greater the risk.	Agile provides guidelines in the form of the Agile Manifesto and its 12 Agile Principles. Teams following these guidelines typically deliver higher quality products and deliver a valued product sooner to the product owner/stakeholders. Having the Product Owner (from the business unit/function involved) and development team work face-to-face on an almost daily basis using the Scrum framework creates quick feedback loops which reduces risk of delivering the wrong product. Risk is further reduced and mitigated by the Scrum framework, which prescribes 2 – 4 week Sprints, capturing detailed requirements only when a User Story is planned into a Sprint, and small, incremental delivery of these Stories. Thus, Agile/Scrum breaks the project down into manageable chunks, which means that the amount of not useful work is small compared to the traditional waterfall method in the case the project is drastically altered or even terminated.
Team Size – multiple implementers, staffing levels, large teams.	Scrum teams are about five-eight members. An emerging Agile operational model, the SAFe® Agile Framework (Scaled Agile, Inc, 2016), provides a comprehensive way to manage and coordinate multiple scrum teams, which can all work on different parts of a large project.
Team diversity – team turnover, cross functional teams, teams not working together in the past.	Scrum teams naturally have some turnover, but in general, they stay together, across projects. This continuity leads to increasing productivity over time and greater accuracy in estimating how long Stories will take to complete.
Uncertainty – technological complexity, technological change, technological newness, team changes	Scrum is specifically designed to deal with change by generating a minimal viable project in short time Sprints (e.g., two-week increments).

Table 1 – Risk Factors related to Project Characteristics

Risk Factors in Project Initiation	**Applicable Agile Practices**
Improper feasibility analysis or business case	The Scrum framework doesn't prescribe a specific project management methodology. Miami (OH) University has adopted the use of what they call Pre-planning the project. This

	step helps capture high level information about the project for the development team members stay focused on their current work. After Pre-planning, the first Sprint of the project, referred to as Sprint 0, delves into the high-level information with the whole Scrum team. Completing these steps compensates for any lack of information collected in the project request. Pre-planning encompasses meeting with the business Project Sponsor, Product Owner (from the business), a Project Manager (from IT), and the IT manager of the application. This collects information to complete the Project Charter and the Stakeholder Analysis with communication points. Completion of these documents teases out stakeholders, project participants and their roles, risks, assumptions, timing, and project scope. Thus, Scrum is no different from other methodologies, which require specific Project Initiation documents like a Project Charter. Sprint 0 includes reviewing the Project Charter and creating a shared vision of the project with the Development Team, Scrum Master, and Product Owner. Other activities include high level design, creating the initial Product Backlog, investigating any known risk areas, providing training for members of the Scrum team, and delivery of the highest valued Stories.
Incomplete or inaccurate initial project cost estimate	The applicable Agile Practices listed in the feasibility/business case section also apply here.
No executive signoff or lack of executive commitment	The applicable Agile Practices listed in the feasibility/business case section also apply here.
Poorly defined scope statement	The applicable Agile Practices listed in the feasibility/business case section also apply here.
Unclear or misunderstood scope and objectives	The applicable Agile Practices listed in the feasibility/business case section also apply here.
Incomplete project charter and success criteria	The applicable Agile Practices listed in the feasibility/business case section also apply here.
Inadequate notification of organizational resources and departments of a new project	The applicable Agile Practices listed in the feasibility/business case section also apply here.
Failure to gain user involvement	Product Owner, as a key part of the scrum team, coordinates with end users/stakeholders of the product. The Scrum Master and Project Manager supports the Product Owner in this role.
Lack of effective project management methods	Scrum prescribes no specific project management methodology. As IT functions become better at using the Scrum framework and Scrum roles, and their portfolio management becomes stronger, many found that the project management steps are no longer needed. Nonetheless, regardless of which stage of maturity an IT function has reached, Scrum has become popular in industry because it is effective.
Lack of management support for assurance practices (quality assurance, project assurance)	Agile guidelines used by the Scrum teams naturally increase quality. Miami's IT function currently has a Quality Assurance team which is developing testing methods, practices, and metrics to increase quality. Metrics provide the evidence of their improvements along with areas for future improvement.

Table 2 – Risk Factors in Project Initiation

PMA increases project success and reduces risk by first, carefully identifying risks and then assuring that the proper agile practices have been used and followed to manage that risk. Certain Agile practices tend to increase risk, especially from a traditional audit perspective (PwC, 2015), such as the lack of documentation as espoused in the Agile Manifesto (i.e., working software valued over documentation). However, it is clear to the authors that certain Agile practices, such as the Product Backlog and Burndown Charts, are inherently controls themselves, even if the Scrum teams who use them do not perceive them as such (based on a comment we received from our main contact). Transparency from Burndown Charts is a governance feature because the team cannot hide its daily/weekly activities and productivity, and thus each team can be monitored for key governance objectives: effectiveness (building the right features/Stories) and efficiency (managing its resources well).

CONTRIBUTION AND NEXT STEPS

While IT project failure continues to be a common topic in IS research, there is still a need to ensure successful project outcomes, especially in the context of Agile/Scrum project management that is surely going to be the norm for most of systems development projects within the next five years (if not before). Our goal is to propose an assurance model that can increase the success rate for IT projects by showing how Agile principles and practices inherently address much of the risks that must be managed for the successful delivery of a project. Due to the space limitations and nature of the paper (research in progress), we did not complete the rest of the project phases.

ACKNOWLEDGMENT:

We wish to thank Mary Brooks of Miami University's IT Services for her significant contributions to our understanding of Scrum IT project management as practiced by Miami.

REFERENCES

1. Agile Manifesto (n.d.) Manifesto for Agile Software Development, retrieved from http://agilemanifesto.org/, retrieved on February 23, 2018.
2. Cagliano A.C., Grimaldi S., and Rafele C. (2015). Choosing project risk management techniques. A theoretical framework. In: JOURNAL OF RISK RESEARCH, vol. 18 n. 2, pp. 232-248.
3. Carbone, T.A., and D.D. Tippett. (2004). Project risk management using the project risk FMEA. Engineering Management Journal 16, no. 4: 28-35.
4. Charette, R. N. (2005) "Why Software Fails," *IEEE Spectrum*, 42 2005.
5. Glass, R. L. (2006) "Looking into the challenges of complex IT projects," *Communications of the ACM*, 49(11):15-18, 2006
6. Hardy-Vallee, B. (2012) *The cost of bad project management.* Gallup Business Journal 2012; retrieved from http://www.gallup.com/businessjournal/152429/cost-bad-management.aspx#.
7. Highsmith, J.A. (2000). Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, 2000 New York: Dorset House, 392pp.
8. Khazanchi, D. and Owens, D. (2010). Project Management Assurance. Available at SSRN: https://ssrn.com/abstract=2456371 or http://dx.doi.org/10.2139/ssrn.2456371.
9. Owens, D. and Khazanchi, D. (2017), "Project Management Assurance: Research in Progress". *MWAIS 2017 Proceedings*. https://aisel.aisnet.org/mwais2017/10.
10. Owens D. M. and Khazanchi D. (2009). Software Quality Assurance in Kidd, T.T. (Ed.), *Handbook of Research on Technology Project Management* (pp. 242-260). Planning and Operations.
11. PwC (2015). "Internal audit: Thinking differently in an agile organization," PricewaterhouseCoopers LLP.
12. Scaled Agile, Inc (2016). SAFe® 4.0 Introduction: Overview of the Scaled Agile Framework® for Lean Software and Systems Engineering, White Paper, July 2016.
13. Scrum Alliance (n.d.). Scrum Alliance, retrieved from https://www.scrumalliance.org, retrieved on February 27, 2018.
14. Scrum Methodology (n.d.). Scrum Methodology, retrieved from http://scrummethodology.com/, retrieved on February 23, 2018