MWAIS 2017 Proceedings

Midwest (MWAIS)

6-2017

Governance Practices over Agile Systems Development Projects: A Research Agenda

Jeff Merhout
Miami University, jmerhout@miamioh.edu

Mary Kovach
Miami University, kovachm2@MiamiOH.edu

Recommended Citation

Merhout, Jeff and Kovach, Mary, "Governance Practices over Agile Systems Development Projects: A Research Agenda" (2017). *MWAIS 2017 Proceedings*. 34.
http://aisel.aisnet.org/mwais2017/34

Governance Practices over Agile Systems Development Projects: A Research Agenda

Jeffrey W. Merhout
Miami University
jmerhout@MiamiOH.edu

Mary Kovach
Miami University
kovachm2@MiamiOH.edu

ABSTRACT

Agile systems development practices are transforming the business of creating and maintaining new information systems. This research-in-progress proposal addresses a key question of whether such Agile practices provide a sound basis for adequate IT governance.

Keywords

Systems development, project management, Agile practices, Scrum, Kanban, SAFe®, governance.

INTRODUCTION

Although practitioners have worked hard to develop methodologies to better manage them, information systems (IS) development has had a long history of failed projects. As a result the information technology (IT) industry has been evolving into the use of so-called Agile project management and systems development practices, such as Scrum and Kanban, and even hybrid versions of these, such as Srumban. These Agile practices are growing in use by industry. For example, a study by Protiviti (2016) found that 44% of companies overall, including 58% of technology companies and 53% of consumer products and retail, are investing in and adopting these practices. Thus, it is safe to conclude that Agile will continue to be a standard practice for a significant percentage of IT functions.

We are in the preliminary stages of developing a better understanding of Agile industry practices and developing a research agenda that we hope will inform our IS research community about the key issues regarding the evolution away from traditional system development lifecycle (SDLC) methodologies, such as the waterfall, to Agile. The main reasons that practitioners tout for adopting Agile practices are that value is developed quicker and risks are better managed because of much shorter development cycles than traditional methodologies that could take many months or even years to deliver a system, which often results in systems that are outdated or even obsolete before they are ever used. It is pretty obvious that software developers, whether in companies that work for clients or develop products for in-house use (such as consumer products companies), would not be continuing to replace their traditional practices with Agile if they were not finding true value from doing so. However, it appears that IT governance research (which will always be a critical part of our discipline) over such Agile practices is not yet a popular topic in the IT literature.

With this research-in-progress overview we propose to develop frameworks and analyses that we can work on as a long-term research stream. For the conference in May, we plan to provide an overview of Agile practices that we have encountered in our many industry partners, which range from a Fortune 25 grocery chain to an industry leader in processing payment transactions for retailers. We also plan to review the results of analyses we have completed as of that time about how traditional IT governance practices are being applied in an Agile setting.

IT GOVERNANCE

IT governance is a broad concept that seems to convey different meanings to different audiences, but some of the terms that generally fall under this umbrella include strategy, compliance, risk management, security and fraud prevention, performance management, quality, and systems documentation. Obviously the level of the organization involved will vary for some of these terms, such as strategic considerations versus documentation, which is usually not strategic at all. From a strategic standpoint, IT governance includes the processes that "deal with the stakeholder governance objectives—value delivery, risk optimization and resource optimization—and include practices and activities aimed at evaluating strategic options, providing direction to IT and monitoring the outcome" (ISACA, 2012, p. 23). Strategic requirements dictate that IT assets be treated as other capital assets and thus subject to systematic processes to approve new acquisitions to provide business value and to guide the development process (i.e., systems development). Compliance with external requirements, such as governmental laws and industry requirements is also a necessary component of governance if an organization is to meet its obligations.

Moreover, IT risk management to ensure that threats to the enterprise are addressed so that IT resources are available as required and information asset vulnerabilities are secured from both physical and logical dimensions is another vital part, as are controls to prevent computer fraud, where information assets can be both the target and the means of attack. At somewhat more of an operational level, new IT systems must have quality (e.g., requirements are met, adequately documented, contain minimal bugs, be secure and reliable), and systems performance must be held to a high standard. In addition, stakeholder management is a vital component of governance where the key stakeholders, such as the business units and functional areas, plus senior management are provided timely and relevant information so they can perform their vital roles in governance, which by nature are interdependent.

RESEARCH AGENDA

Although all of the above are indeed critical components of overall IT governance, we can only hope to address a small subset for this first phase of our long-term research agenda. Thus we propose to map a traditional set of SDLC phases (e.g., system definition, analysis, design, development, quality, testing, implementation, and maintenance) to Agile practices, such as Scrum, Kanban, and Scrumban, and to identify governance points and determine how Agile does/does not address them. In other words, we hope to create a gap analysis between ideal governance practices and what Agile is capable of providing without additional practices. For example, documentation may or may not be an issue with Agile, but the popular press tends to suggest that Agile practices shortchange documentation, perhaps because the Agile Manifesto states that “working software” is valued more than “comprehensive documentation” (LeanDog, 2015).

We believe our long-term research stream will provide insights to other IS researchers about some of the key questions that are related to the evolution to Agile development and perhaps even to practitioners about possible governance gaps that might exist. And, as a minimum, the knowledge we gain will be informative for our IS students who are increasingly likely to enter an IT workgroup that is already practicing Agile. Accordingly, for the short term, we propose the following high-level research questions:

- Do Agile system development lifecycle (SDLC) practices, in general, provide an adequate framework for governance?
- Does a specific emerging Agile operational model, SAFe®, provide an adequate framework for governance?

MAPPING AGILE TO TRADITIONAL SDLC

Although there are many varieties of the traditional system development lifecycle, for purposes of this analysis, we will use a very generic version (of which it is likely no actual company uses) as a starting point for comparison to different Agile practices: system definition, analysis, design, development, quality, testing, implementation, and maintenance.

Example 1 – A comparison of the traditional SDLC to the SAFe® Agile Framework (note that XYZ Corp is a pseudonym for a real company using SAFe):

Traditional SDLC phase	**Typical governance actions/artifacts**	**Comparable Agile practice (SAFe® per Scaled Agile, Inc., 2016;** as adapted by XYZ Corp.**)**	**Agile governance actions/artifacts**

System Planning/ Project Definition	Goals & scope; feasibility analyses; project plan; project team	Product Manager Generates Ideas and Manages the Program Backlog (Intake);
		Epic Backlog
		Program Increments (PI) last approximately 8-12 weeks (will vary by firm)
			Portfolio Review Committee;
			Product/Solution vision;
			Enterprise Quarterly Planning Increment (e.g., 8-12 weeks);
			Enterprise Architect;
			Concept Review Board (CIO, CTO, CSO)
			• Aligned with strategy?
			• Acceptable level of risk and technical

			debt?
Analysis	Approved user requirements	Epic: multiple Features over 2-3 Program Increments (PI); Features/Stories	Program Backlog; Product Owner
Design	Conceptual design; Detailed design specifications; Test plan	Agile Release Train (ART) - SAFe teams use a variety of Agile practices, e.g., Scrum, XP, Kanban	Release Train Engineer (RTE); DevOps; Design Review • Aligned with Standards (Interoperability, Security, and Reliability)? • Feasible to implement and operate?

Development/Build/Coding	Software programming documentation	Execution
		Sprints/Iterations (e.g., 2 weeks) by ARTs
			Scrum Master;
			Per iteration: Iteration Planning, Daily Stand-up, Backlog Refinement, Sprint Backlog, Team Demo of Shippable Product, Sprint Burndown Chart, Iteration Retrospective;
			Per PI: Scrum of Scrums, PO Sync, Release Management Meetings, System Demo, Inspect & Adapt workshop;
			Operational Readiness Review
			• Operationally ready to support solution?
			• Built what was approved?

Quality/ Testing	Test cases; Unit testing; Integration testing; UAT	QA Tests	Inspect & Adapt workshop
Implementation/Conversion	Implementation checklist	Release as needed per market demands, such as end of PI or end of Iteration	DevOps
Maintenance	Change request documentation	Program Backlog (Intake)	Varies, depending on scale

Table 1. Comparison of Traditional SDLC to SAFe®

Example 2 - A comparison of the traditional SDLC to the Adaptive Project Framework (Wysocki, 2009):

Traditional SDLC phase	**Typical governance actions/artifacts**	**Comparable Agile practice (e.g., Adaptive Project Framework;** Wysocki, 2009**)**	**Agile governance actions/artifacts**
System Planning/ Project Definition	Goals & scope; feasibility analyses; project plan; project team	Version Scope: StageGate 1 Approval	StageGate 1 Approval
Analysis	Approved user requirements	Version Scope: StageGate 2 Approval	StageGate 2 Approval
Design	Conceptual design; Detailed design specifications; Test plan	Not evident from source analyzed (i.e., Wysocki, 2009)	Not evident from source analyzed (i.e., Wysocki, 2009)
Development/Build/Coding	Software programming documentation	For numerous Cycles: Cycle Plan; Cycle Build: StageGate 3 Approval	Each Cycle StageGate 3 Approval
Quality/ Testing	Test cases; Unit testing; Integration testing; UAT	Client Checkpoint: StageGate 4 Approval;	Each Cycle StageGate 4 Approval; Close Project – Post-version review
Implementation/Conversion	Implementation checklist	Not evident from source analyzed (i.e., Wysocki, 2009)	Not evident from source analyzed (i.e., Wysocki, 2009)
Maintenance	Change request documentation	Not evident from source analyzed (i.e., Wysocki, 2009)	Not evident from source analyzed (i.e., Wysocki, 2009)

Table 2. A Comparison of Traditional SDLC to SAFe

CONCLUSION

Based on our early analyses, it seems apparent that one particular Agile framework, SAFe®, is more of an SDLC model than one focused on governance (a conclusion confirmed by conversation with an enterprise architect at XYZ Corporation). However, good governance is a managerial responsibility regardless of the methodology used. Although there might be some gaps in the governance of documentation during the design phase of the Agile methodology (as practiced within the guidelines of the SAFe framework), we have not identified any significant deficiencies in the overall governance practices at XYZ Corporation. Thus, we conclude (at least for this early stage of our research) that Agile is no more problematic with regards to governance than traditional SDLC methodologies (such as waterfall) – it is the responsibility of management (and all IT staff) to practice good governance. Frameworks are simply tools, and good management should utilize the tool appropriately in in the context of the strategic and practical requirements and restrictions of the system being developed.

ACKNOWLEDGMENTS

We thank all authors, committee members, and volunteers for their hard work and contributions to the conference, and we especially thank personnel at XYZ Corporation for their generous help (e.g.., their time) as we studied this topic.

REFERENCES

1. LeanDog, Inc. (2015) Agile Discussion Guide, version 3.2. Cleveland, OH: LeanDog, Inc.
2. ISACA (2012) *COBIT® 5: Enabling Processes*, Rolling Meadows, IL.
3. Protiviti (2016) From Cloud, Mobile, Social, IoT and Analytics to Digitization and Cybersecurity: Benchmarking Priorities for Today's Technology Leaders, accessed on February 06, 2017 from https://www.knowledgeleader.com/KnowledgeLeader/Content.nsf/Web+Content/SRFromCloudMobileSocialIoTandAnalytics!OpenDocument
4. Scaled Agile, Inc (2016) SAFe® 4.0 Introduction: Overview of the Scaled Agile Framework® for Lean Software and Systems Engineering, White Paper, July 2016.
5. Wysocki, Robert K. (2009) Effective Project Management: Traditional, Agile, Extreme, 5th Edition, Indianapolis: Wiley.