Can Agile, Lean and ITIL Coexist?

Aaron W. Smith, Volteo, Phoenix, AZ, USA
Nayem Rahman, Portland State University, Portland, OR, USA

ABSTRACT

Information Technology Infrastructure Library (ITIL) is frequently perceived as overly wordy and bureaucratic while Agile and Lean methodologies are often positioned as a rejection of heavy handed process and governance. But IT Service Management, Agile, and Lean are all focused on the same thing: How to get valuable work done quickly and efficiently in the complex world of IT in order to enable a business’s competitive edge. Delivering IT solutions at the speed-of-business while maintaining quality, performance and availability of the Live Environment is a constant balancing act among simultaneous constraints. ITIL, Agile and Lean all contain best-practices to define and manage these constraints and the contexts in which they exist. The authors will compare and contrast the core of these methodologies to show how they align with the goal of providing a roadmap for successful implementation and coexistence.

KEYWORDS

Agile, Flow, Information Technology, IT, ITIL, Lean, Workflow

1. INTRODUCTION

A simple Google search for “Agile vs. ITIL” will net over 150 thousand results and “ITIL vs. Lean” well over a half-million more. “ITIL bureaucracy” finds over 400 thousand results. These results will of course contain defenders of ITIL (Information Technology Infrastructure Library) arguing that it is not bureaucratic, but even supporting articles exist due to an overwhelming perception that it is true. To underscore the point, adding the results for “Agile vs. bureaucracy” with “Lean vs. bureaucracy” gets to just short of two million results.

IT is consistently under pressured to deliver more capability with less resources so working “smarter not harder” has never been more necessary. Choosing to leverage industry Best Known Methods (BKMs) and thereby learning from others’ successes and failures without incurring the cost and risk directly is a wise technique. IT teams of any size cannot afford to turn away from tools and knowledge (Sarabia et al., 2015; Sharma and Singh, 2015) which may help them survive and succeed in the face of current and looming challenges.

A bottleneck or ‘weak-link’ in the value-chain to deliver a product or service not only slows the throughput but can in-effect cancel out investments or improvements made in other ‘links’ of the chain. In today’s environment, IT is a critical component in delivering all aspects of business including sales, marketing, manufacturing, HR and finance. Decision makers must include and align IT to the business’ goals whenever attempting to optimize a research, development, manufacturing, marketing, delivery or support workflow.

As companies move to compete in lines of business where the margins are ever tighter, the ‘speed of business’ increases significantly, and the risks in failing to maximize investments are greater; companies cannot afford to leave any opportunity to better align workflow and lean-out processes.
In this article we will look briefly into the history and core tenets of each of these three methodologies, call out differences and possible conflicts as well as the similarities and points of connection. We will include one example of how the ITIL, Agile and Lean processes may not only coexist but strengthen each other to create exponential returns on an investment in their adoption.

2. LITERATURE REVIEW AND THEORETICAL FRAMEWORK

IT departments need to deliver business value faster. Traditional IT projects that follow waterfall method may take six months to a year to complete a software project (Rahman et al. 2011). Taking longer time before realizing business value is considered a major limitation and inefficiency of the waterfall method. Technologies used in providing IT capabilities change very fast. With the waterfall method, business organizations may find technology and applications delivered by IT projects to be old, inefficient or no longer serving the business purpose. To overcome the limitations of waterfall method serving projects, IT organizations and developer communities have come up with agile and lean software development methodologies.

Agile and Lean software development methodologies emerged early in the 21st century. When IT projects are implemented within a short time, they provide customers with faster turnaround between idea and solution. Waterfall has been known to take such a long time the IT solution delivered had become obsolete before the project was implemented. Agile Development methodology came into picture to overcome the shortcomings and inefficiencies of traditional waterfall method (Abbas et al. 2008; Sommerville, 2010). Among the agile methods the most popular ones are Extreme Programming (XP), Scrum and Lean software development (Campanelli and Parreiras, 2015; Reliable Software Resources Inc. 2011).

Agile methodologies have certain shared characteristics including frequent delivery of projects via multiple iterations; rigorous testing of software (Afzal et al., 2016), daily stand up meetings; collaborative work among team members; direct communication link between developers and customers (Keil and Carmel, 1995); and working software over documentation (Abbas et al. 2008). IT practitioners have identified several advantages of agile software development which include greater agility, realistic customer expectations, a more productive development team, early warnings of problems, and more visible and practical milestones (Koch, 2011). In Agile development, active customer participation throughout the software development lifecycle allows for keeping them up-to-date on progress and problems and meeting their evolving needs and expectations. Customers are able to provide developers early feedback to course-correct, avoiding major rework. Team members’ Daily Stand-Up (DSU) meeting and collaboration helps getting better insights in the development processes (Petersen and Wohlin, 2008; Chappell, 2012). Agile methodologies have a few issues as well. The iterative nature of Agile software development requires frequent release of require functional products to production or market. This causes management overhead; higher number of coordination and communication tasks, and increased maintenance effort since many different versions need to be supported (Turk et al. 2000; Petersen and Wohlin, 2008).

“Lean initiatives in manufacturing, logistics, services, and product development have led to dramatic improvements in cost, quality and delivery time” (Poppendieck, 2007). Lean software development methodology is meant for writing quality software rapidly, repeated, and reliably. Toyota has been successful in using lean software engineering principles for its vehicles (Poppendieck, 2007). IBM has come up with 18 best practices for lean development governance. These practices support such modern approaches to development as Lean Software Development and Scrum (Ambler and Kroll, 2009).
Computer Mediated Interorganizational Knowledge Sharing: Insights from a Virtual Team Innovating, Using a Collaborative Tool
www.igi-global.com/chapter/computer-mediated-interorganizational-knowledge-sharing/54255?camid=4v1a

An Agile Project System Dynamics Simulation Model
www.igi-global.com/article/an-agile-project-system-dynamics-simulation-model/109090?camid=4v1a