



Contents lists available at ScienceDirect

Government Information Quarterly

journal homepage: www.elsevier.com/locate/govinf

Agile innovation management in government: A research agenda

Ines Mergel Dr.

University of Konstanz, Germany

ARTICLE INFO

Article history:

Received 16 July 2016

Received in revised form 19 July 2016

Accepted 19 July 2016

Available online xxx

Keywords:

Agile government

Agility

Agile development

Digital service delivery

Innovation

ABSTRACT

Governments are facing an information technology upgrade and legacy problem: outdated systems and acquisition processes are resulting in high-risk technology projects that are either over budget or behind schedule. Recent catastrophic technology failures, such as the failed launch of the politically contested online marketplace Healthcare.gov in the U.S. were attributed to an overreliance on external technology contractors and failures to manage large-scale technology contracts in government. As a response, agile software development and modular acquisition approaches, new independent organizational units equipped with fast reacting teams, in combination with a series of policy changes are developed to address the need to innovate digital service delivery in government. This article uses a process tracing approach, as well as initial qualitative interviews with a subset of executives and agency-level digital services members to provide an overview of the existing policies and implementation approaches toward an agile innovation management approach. The article then provides a research framework including research questions that provide guidance for future research on the managerial implementation considerations necessary to scale up the initial efforts and move toward a collaborative and agile innovation management approach in government.

© 2016 Published by Elsevier Inc.

1. Introduction

Previous waves of digital innovation management were closely connected to fads and fashions in public management. The New Public Management era for example brought about disaggregation, competition, and incentives to outsource digital service delivery and reduced skills and capacities in government (Dunleavy, Margetts, Bastow, & Tinkler, 2005). The result is that contract managers in government are oftentimes following a performance-based acquisition process that aims to anticipate the final software products within a rigid framework of contract fulfillment obligations. These IT acquisition norms led to increased complexities, delays or even failed delivery of digital services (Anthopoulos, Reddick, Giannakidou, & Mavridis, 2016): as an example the U.S. Healthcare.gov virtual marketplace to sell mandatory health insurance coverage to citizens failed to launch and had to be rescued by a team of Google engineers in an emergency turn-around that cost several 100 million dollars over the initially contracted price. As a result, practitioners and researchers are now calling for adaptive (Janssen & van der Voort, 2016), anticipatory (Bertot, Estevez, & Janowski, 2016), and agile (Balter, 2011; Margetts & Dunleavy, 2013) approaches to reintegrate digital service delivery with a holistic focus on human- and client-centered design delivered through shorter development cycles.

Agile innovation management is introduced here as an umbrella term that describes a set of project management and software

development processes, adjusted procurement procedures, combined with HR policies, and organizational and managerial approaches to support innovative digital service delivery in government. Innovation in government software development are created by using an agile software development approach adopted from private sector and especially IT sector organizations that are involved in creating software projects on a shortened project management life cycle. In comparison to the traditional waterfall project management approach in which each phase sequentially follows the previous phase, an agile approach focuses on shorter development phases and radical collaboration with the client in each phase. Subprojects and results are presented and tested immediately and not delayed until the final product is presented at the end of the fully completed contract. The method evolved as part of a 'new product management' approach in Japan and was subsequently adopted by the IT industry (Takeuchi & Nonaka, 1986).

Recent developments have shown that government agencies are implementing similar approaches in order to update large-scale legacy system and adapt to environmental changes and citizen requests faster. Driven by negative experiences, the overreliance on external IT contractors, and management oversight failures, the current administration took the introduction of HealthCare.gov as a stepping stone to introduce agile development processes. In 2013, HealthCare.gov was initiated by a top-down Presidential mandate to create an online marketplace that brings together insurance providers offering health insurance to individual citizens based on the income, family status, and location of their residence by state. The implementation was a massive IT failure that included over 50 subcontractors and IT spending of over \$800

E-mail address: Ines.mergel@uni-konstanz.de.

million (see for example Christy, 2016). The contractors and those responsible at the Department of Health and Human Services never conducted test runs with a subset of user groups, instead the decision was to go live on the day of the reveal of the platform. For the administration this was the first time they saw the platform running after months of development. Similarly other high-risk IT projects fail in government: 94% of IT projects in the U.S. federal government are over budget and behind schedule and 40% are never finished (Torgovnick, 2016).

The traditional waterfall software development methods in the current acquisition paradigm of government contracting is highly intransparent and dissatisfying both for government clients who do not receive the expected products, as well as for contractors who are tied to the existing acquisition and contracting rules to deliver what is defined in incomplete IT contracts. As Read (2016) in a recent TechInsider post said: “Government workers tend not to invite the customers to see the sausage being made, but wait until the silver platter is ready”.

The data collected for this study were informed by an initial literature review of mostly the computer science literature to derive the core concept of agile software development and to contrast it to traditional development methods. This distinction drawn in the literature as well as in government policies, guidance documents, and reports were then used to inform a semi-structured interview outline for government officials in the U.S. federal government (Drever, 1995). The interview partners included eight top managers of the central digital transformation team located in the General Services Administration (GSA) responsible for replicating practices across the federal government and representatives of five different federal departments in the U.S. government which have already started to apply agile innovation management approaches. The selection includes one case only – the U.S. federal government – mainly because the case is well-documented by government technology media articles, but also because each agency faces similar contextual opportunities and constraints (Strauss & Corbin, 1998). Other governments, such as the United Kingdom or the Netherlands have gained similar experiences and more research is needed to understand each case in depth. The interviews, document search and tracing, as well as the existing literature were then used to draw initial conclusions about the concept of agile innovation management.

This article first reviews the development of agile approaches, the underlying principles, the components of an agile development process in contrast to a traditional waterfall project management approach preferred by IT contractors, and then highlights the benefits and challenges of agile development in government. The article then presents insights from a process tracing approach (Collier, 2011) and initial qualitative interviews and presents a two-layered research framework using agile principles for the implementation of agile innovation management based on the insights of the U.S. federal government case. Agile innovation management is presented here as a comprehensive framework that highlights how agile methods also need policy and management changes in order to contribute to government innovations. Finally, the article ends with a set of open research questions that need empirical evidence to understand the concept of agile government, acquisition processes, cultural changes, as well as HR and training needs.

2. Agile development process: from sequential to overlapping software development

The development, management, and operation of IT projects in government traditionally follow a waterfall approach: tackling one piece of the development phase at a time and providing the final product to the buyer. This is largely attributed to the current acquisition procedures and contracting practices. Recently, more and more agile development approaches that are well-established in the private sector have been moved into government operations, challenging project management, acquisition rules and standards, as well as the culture of project teams and contract managers. In the following, a brief overview of these two different approaches is provided.

2.1. Traditional software development process: sequential, waterfall software development

Most IT contracts and internal modes of software development in the public sector use a sequential process, in which one phase has to be completed before the team is allowed to progress to the next phase. This progression “flows from top to bottom, like a cascading waterfall” and is therefore called the waterfall development method (Royce, 1970). The core belief here is that by finishing each phase and eliminating any possible mistakes for this phase, future phases won't be impacted by mistakes and the project team won't lose time and money by going back to fix the mistakes.

The downside is that contract managers and developers must have a fully developed plan before they write a request for proposals, sign a contract, and start work on the final product. The reality in government is however, that IT professionals and contract managers don't have all the final details available to define and specify the details of a contract. This circumstance oftentimes leads to contract add-ons or extensions to accommodate for changing internal needs or to fill in the gaps that exist at the beginning of the request for proposals phase.

Given the fixed structure of IT contracts in government, risk-aversion to veer outside the contractual obligations and oftentimes unforeseen adjustments, government organizations tend to follow initial rigid contractual structure (Balter, 2011). Project phases are predefined with deadlines and deliverables that are tied to payments and leave little room for fluid adjustments that might be necessary to fit in initial omissions or client changes along the development process. As a result, waterfall methods are criticized for their rigidity, inflexibility, and lack of communication with the clients and users during the development process. They tend to fill the contract requirements and support risk-averse approaches of government contract managers, but might not satisfy the users, clients, or government employees who have to use the final product to support the mission of their organization (GAO, 2012). At the end, waterfall methods are not able to respond to changes in the environment that are destined to happen, especially in large-scale, long-term delivery contracts. As a result, expensive follow-up service contracts have to be signed to manage required changes and respond to the actual user requirements. A waterfall process shown in the following figure therefore favors the final product and contract compliance over users' needs and actual use of the final product (Fig. 1):

A major disadvantage of a waterfall development approach is that while the project team works through each phase, it can never respond to changing needs and requirements and never knows the true progress or final outcomes. Only at the end – at what are usually mega launches – the software might fail and problems are not known until the end. The customer or contract owner is only invited to get involved in testing the final product at the end of the development process. Evaluations of the project status do not occur until the very end.

2.2. The alternative: agile, overlapping development

While the step-wise process initially made sense, Royce (1970) stated already in 1970 that it is “risky and invites failure”. He continues to

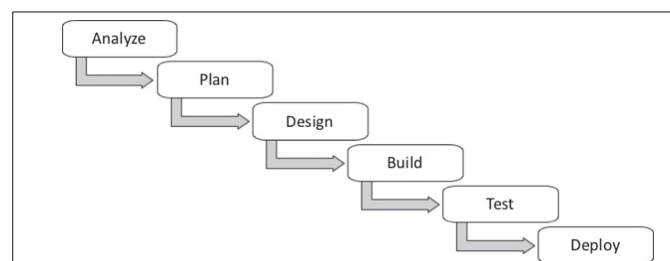


Fig. 1. Waterfall development process.

Download English Version:

<https://daneshyari.com/en/article/5110646>

Download Persian Version:

<https://daneshyari.com/article/5110646>

[Daneshyari.com](https://daneshyari.com)