



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Governments are facing an information technology upgrade and legacy problem: outdated systems and acquisition processes are resulting in high-risk technology projects that are either over budget or behind schedule.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.

<http://dx.doi.org/10.1016/j.giq.2016.07.004>
0740-624X/© 2016 Published by Elsevier Inc.

Please cite this article as: Mergel, I., Agile innovation management in government: A research agenda, *Government Information Quarterly* (2016), <http://dx.doi.org/10.1016/j.giq.2016.07.004>

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

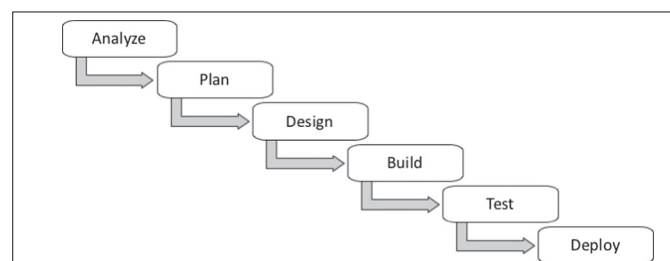


Fig. 1. Waterfall development process.

explain: “The testing phase which occurs at the end of the development cycle is the first event for which timing, storage, input/output transfers, etc., are experienced as distinguished from analyzed.” (Royce 1970:329). The reality in government is that there are continuously changing developments and developers cannot assume that agencies, especially non-IT contract managers, are able to adequately predict the specifications of large-scale IT projects. This is an immense challenge for the acquisition process and the way vendors and contractors who win a bid are then tied to the process and predefined outcomes.

In contrast to waterfall development methods, agile development is a method that involves creating, testing, and improving technology products incrementally, instead of waiting for the foolproof delivery by the end of the contract period in a traditional IT contracting agreement. Agile methods were first introduced outside the IT industry, but are mostly known for its advances in software development: the physicist and statistician Walter Shewhart of Bell Labs began applying Plan-Do-Study-Act (PDSA) cycles to the improvement of products and processes (Goldman & Preiss, 1994). His approach constitutes the first iterative and incremental development approach to rapidly respond to changes and actual findings in the development process.

In 1986, Takeuchi and Nonaka introduced a team-oriented approach to bring products to market faster and called it the “rugby” approach “where a team tries to go the whole distance as a unit, passing the ball back and forth” (Rigby, Sutherland, & Takeuchi, 2016b; Takeuchi & Nonaka, 1986). They started to consider large software development projects as multiple independent tasks. Takeuchi and Nonaka (1986) called their iterative and incremental agile software development framework the *scrum* method: small, self-organizing, adaptive teams that are tasked with solving complex problems using overlapping development phases. The tasks are broken down into small, manageable modules and use the short development cycles, hold brief daily “stand-up” meetings to review progress and identify roadblocks that might prevent them from moving forward. In addition, progress is visualized in a process management system called *Kanban*, that helps teams to reduce lead times and the amount of work in process. In the meantime, these principles are also included in the agile development manifesto (Agile Manifesto, 2001).

The starting point for an agile development process is qualitative research conducted by the project team to understand what the needs of the clients are. Clients in government can be agencies and contract managers themselves, or external clients, such as partners or citizens who will use the final product. These user stories are collected using the language of the clients to avoid misconception or translation mistakes with the hope that using language that is known and familiar to the client will ultimately make it easier for them to use the final product and easily integrate it into their day-to-day use. An important principle of agile development processes is to be able to react to changes that are initiated to the changing requirements from clients, or as Cockburn (2002, 10) puts it, it is “a declaration of prioritizing for project management vulnerability with respect to shifting requirements, shifting technology, and a shifting understanding the situation.”

The agile development method follows so-called ‘sprint cycles’ so that the development teams by design fail often and early, instead of failing catastrophically and wasting taxpayer money as seen with the U.S. online marketplace for sale of health insurances, *Healthcare.gov*. The iterations – or sprints – are following the traditional waterfall in each segment, but usually take between 1 and 8 weeks for each segment, and projects lasts no longer than six months. Similar to project teams in the IT industry, this approach follows the motto: ‘fail fast, early, and often’ with small prototypes that rapidly prove (or disprove) an idea before much is invested (Leetaru, 2016). The initial plan serves as guidance, but the plans are revised after each iteration and the development output is demonstrated to the client and other stakeholders. This early evaluation phase reveals which requirements may not have been fully addressed and can be added as a work package for the next sprint. One important characteristic of an agile approach is therefore

radical collaboration: clients are working closely with IT engineers and developers.

Another important characteristic of an agile approach is continuous delivery of output to the clients (Agile Manifesto, 2001; Martin, 2003): the fluid involvement of the client in the development process allows for continuous feedback and improvement in all stages of the process. This requires an increased time commitment by the client – be it a government agency’s representatives or the citizens. These agile characteristics shift practices and values: the value is not generated by delivering a final product that fulfills the initial RFP or contract. Instead, value is created by putting the client at the center of the project and delivering a final product that benefits the client and creates value for the public. Similar modular software development approaches include rapid or spiral development as alternatives to waterfall methods. Here – as defined by the case selected – I focus on agile approaches only.

The following figure shows the iterative sprint cycles as part of the overall development process (Fig. 2):

Rigby, Sutherland, and Takeuchi (2016a) note that agile methods can be best applied in contexts with complex problems when the final solution might not be predictable or is initially unknown to both clients and developers – especially environments like government. They also note, that agile methods need training and potentially behavioral change, not just in the use of new technologies and approaches, but also in terms of attitudes toward contracting and involvement of clients and developers in project teams (Agile Manifesto, 2001).

Using an agile approach supports more adaptive governance approaches as promoted by Janssen and van der Voort (2016): instead of predicting and prescribing the outcomes, an adaptive government can use agile methods to increase its flexibility in responding to changes and might in the process even deliver less costly innovations in government. However only using agile software development approaches is not enough to fundamentally change how government can tackle the update and legacy problems it faces. Instead, a multi-layered approach is needed that goes beyond the development approach. In the following, based on the case of the U.S. federal government a comprehensive agile innovation management approach is outlined.

3. Implementing agile innovation management practices: the case of the U.S. federal government

Tracing the process of agile innovation management in the U.S. federal government is relatively difficult, because there is no official mandate for a comprehensive strategy at this point. Individual agencies are making progress using initial guidance from the Office of Management and Budget (OMB) to adapt their IT acquisition practices or collaborate with the General Service Administration’s technology transformation service 18F¹ to get advice on how to use agile methods. Agencies were asked to submit confidential budgets that are not publicly available to outline the need for resources to invest in agile innovation management practices. In the following, Collier’s (2011) approach is used to trace the process through which government documents has encouraged agencies to experiment with agile methodologies and improve their acquisition practices. The goal is to make causal observations by combing document reviews with empirical evidence from initial qualitative interviews.

A recent GAO report (2015) has criticized the existing standard operating procedures for IT acquisition: “Federal investments in information technology (IT) have often resulted in multimillion dollar cost overruns and years-long schedule delays, with questionable mission-related achievements.” A case in point is the *HealthCare.gov* implementation disaster. GAO (2014) has attributed the failure of the IT platform to

¹ The agency name 18F is derived from its location at 18th and F Street in Washington, DC.

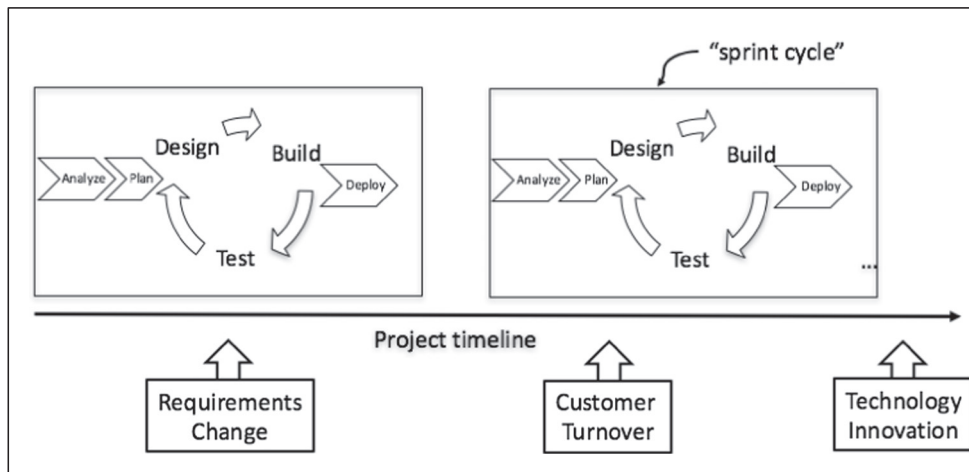


Fig. 2. Agile sprint cycles.

mistakes in contract planning and lack of oversight practices among the responsible Department of Health and Human Service's IT staff.

The Office of Management and Budget (OMB) has instructed federal departments and agencies to use part of their IT budget to set up digital service teams modeled after the U.S. Digital Service team – a quick response team located in the White House that helped with the recovery of HealthCare.gov. This shift in moving resources toward improving customer-centric service delivery follows the 2011 Executive Order 13571 (The White House, 2011) and OMB's implementation instructions (Office of Management and Budget, 2011), the 2012 White House' Digital Strategy (The White House, 2012b), and the Presidential Memorandum instructing the federal government to work toward 'Building a 21st Century Digital Government' (The White House, 2012c).

In 2014, the White House established the U.S. Digital Services team – a quick response team of IT engineers recruited from the private sector to help with Presidential priority IT projects – and the General Services Administration's 18F software development teams. Engineers are recruited from technology companies, including Facebook, Twitter, Google, Microsoft, or Amazon. The so-called 'digital swat teams' of USDS and 18F are paired with public servants using agile software development methods to collaboratively work on high-priority projects, such as the digitization of the U.S. immigration service forms which are still mostly paper-based, or the Department of Veterans' Affairs' education and health service delivery processes.

In addition, the introduction of the Presidential Innovation Fellow program (The White House, 2015) – an HR instrument that allows "Tour of Duty" appointments – are creating opportunities to move external talent from the private sector into government for short periods of time to conduct specific tasks and then return to their previous positions.

Similar to the UK and soon the Italian government (Amazon.com, 2016; UK.gov, 2016), the U.S. federal government has explicitly adopted an agile government approach in 2012 (The White House, 2012b, The White House, 2011). The Obama budget 2016 includes \$106 million to "scale and institutionalize" the still-new U.S. Digital Service by creating similar agency-level digital teams within the largest 25 federal agencies. In some pockets of the U.S. federal government, especially in defense and intelligence agencies, agile methods have been tried and proven to be effective ways to improve software development processes. For example, the U.S. Army has created simulation programs using both agile and traditional methods (Surdu & Parsons, 2006). The Federal Bureau of Investigation (FBI) has worked with its contractor Lockheed Martin on the Sentinel project to create a case management software to change how the Bureau gathers, stores, and links data (Fulgham, Johnson, Crandall, Jackson, & Burrows, 2011). Especially the upfront definition of requirements as part of the standard development process is considered as inefficient and ineffective and oftentimes takes years ([\[2012a\]\(http://2012a\)\), so that the project team evaluated the existing software development methodologies and moved to sprint cycles including outside stakeholders in the process and more effectively managed risks, costs, and outcomes. Small teams of a manageable size \(up to 15 team members\) are set up to create collaborations between FBI employees and government contractors. These teams are using iterative development within the scrum project management framework: iterative development 'sprints', each sprint involves the completion of a small set of tasks in two week cycles, reducing the risk of large scale failure.](http://The White House,</p>
</div>
<div data-bbox=)

At the Department of Defense, the term agile is used as an umbrella term for many different iterative development processes, including Scrum, Lean Development, or extreme programming (XP) (Broadus, 2013). Agile methods are included into contract documents as detailed in the 2010 [Defense Acquisition Bill \(2010\)](http://Defense Acquisition Bill (2010) (Sutherland, 2012)) (Sutherland, 2012), however DoD officials experienced that for example technical documentations were missing at critical design reviews and therefore recommends to invest more in training contract managers and contractors in agile practices (Lapham, 2012). In a recent speech, the Secretary of Defense Carter highlighted the urgency to become more agile as a whole enterprise: "DOD doesn't have many effective ways to harness promising technologies they come up with. We need to fix that. I don't want us to lose out on an innovative idea or capability we need because the Pentagon bureaucracy was too slow to fund something, or we weren't amenable to working with as many startups as we could be." (U.S. DoD, 2015).

In addition to these first experiments, 18F has identified the need for an increased knowledge and expertise among their federal clients to change the government's IT acquisition practices. They are working toward improving acquisition and contract officers ability to know what to do and questions to ask at their next acquisition negotiation with contractors (Christy, 2016). Contractors have to agree the open source approaches, so that source code can be reused across the government.

4. Building blocks of an agile innovation management approach

The agile innovation management framework as observed in the U.S. federal government can be divided into two layers: the first layer – or the basis of the framework – includes foundational policies that need to be adapted in order to shift once learned behavior and practices toward an agile practice. The second layer is the management layer that includes management activities focused on innovative processes, and leadership activities focused on providing cover for all the management activities.

4.1. Base layer: policies

The base layer includes innovative HR policies. In the case of the U.S. federal government only one major HR policy change was introduced

during the eight-year presidency of President Obama. He created a new program that is called the Presidential Innovation Fellows Program (The White House, 2015). Under the umbrella of this program, IT talents were recruited “skilled in technology or innovative practices to serve in the Federal Government to work on some of the Nation’s biggest and most pressing challenges.” Among the recruits the Executive Order specified that executives, innovators, or successful entrepreneurs were asked to conduct a so-called “Tour of Duty” and spend short term stay in the federal government in order to create “meaningful solutions that can help save lives and taxpayer money, fuel job creation, and significantly improve how the Federal Government serves the American people.” These time-limited appointments are meant as an incentive to briefly serve the country and especially infuse the federal government with skills and practices from the private sector. In practice, several of the PIFs (Presidential Innovation Fellows) extend their stays to finish their projects beyond the initial six months’ time period or even moved into permanent positions in government.

The second type of policies focuses on updated acquisition policies that allow agencies to use agile methodologies when they write Request for proposals from vendors. As Mark Schwartz, Chief Information Officer of the U.S. Citizenship and Immigration Services publicly stated, it is important for the government to “Buy competent teams, rather than buying a product.” (Read, 2016). Using Agile BPAs (Blanket Purchase Agreements), vendors, who were already preapproved on the agencies schedules, have to showcase prototypes of their final products and agree to deliver their products using agile methodologies with sprint cycles. Agile BPAs were introduced using the Office of Management and Budget’s Modular Contracting Guidance (GAO, 2012; The White House, 2012a). Under this framework the expectation is that government agencies invest in smaller projects and increments and open opportunities for small businesses to compete for government contracts. This will reduce the risk exposure and avoid sunk costs of “grand design” projects.

4.2. Management layer: process management and leadership

The management layer of the agile innovation management framework includes process management and agile leadership – as the umbrella to facilitate all other activities in the framework.

First, the process layer includes a strict application of *agile methodologies*. An important first step is research. The client’s real underlying needs are researched using qualitative exploration methods with focus groups, end-users, and agency staff to understand what products or processes are actually needed. This user- or human-centered design approach steps away from contract management requirements and pushes the end-users of the projects into the focus. They define what products they need and which processes they will actually use. As Robert Read (2016), 18F co-founder and 18F consulting founder suggests, agile teams need to ask only open-ended questions rather than metric survey questions. The research team uses A/B testing, surveys of at most nine users, and in-person interviews. After that a prototype is quickly built and as part of the first sprint cycle, developers and agency partners improve the product iteratively and then test publicly in a so-called “moment-of-truth” phase. Lastly, the fully developed product becomes operational for all end users.

18F teams up with other agencies using their five business lines: 1) develop and build custom products for agencies, 2) develop innovative ways to buy technology, 3) provide platforms that departments and agencies can use, 4) consulting services to agencies to implement their own digital services teams, and 5) provide training in modern digital services techniques. One support function is to provide contract writing support. This so-called “RFP ghost-writing” advice helps agency-level contract managers understand how to include agile methods into RFPs and how to identify suitable vendors and contracts. In case agencies decide to use 18F’s services to build new software, inter-agency agreements are signed on a fee-for-service basis.

At the top of the management layer, *agile government leadership* is necessary to provide so-called “air cover” for all activities from the top. Beyond strategic planning activities, leadership is responsible for the integration of the digital services teams, software developers, and agency-level staff. The expectation is to create hyper transparency in order to promote values and processes, support policies, provide support for teams to experiment with new methods, or outreach. Moreover, leadership is necessary to protect the team culture and provide incentives for external IT talents. The following figure summarizes the policy and management layer of the agile innovation management framework (Fig. 3):

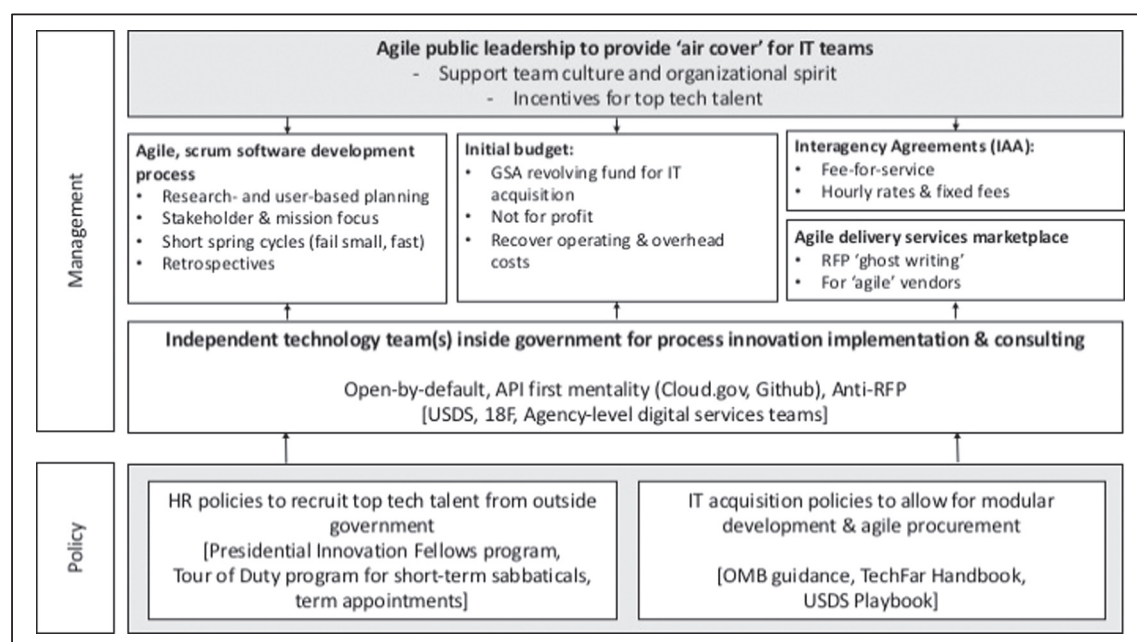


Fig. 3. Agile innovation management building blocks.

5. Discussion and future research

Building on previous approaches, such as Janssen and van der Voort's (2016) suggestions for an adaptive government, agile innovation management can be considered as a value framework for the overall enterprise of government. It goes beyond the project- or process-based viewpoint and therefore needs to evolve around a set of principles that not only focus on the software development process.

An agile innovation management framework needs to consider a series of agile innovation principles that have the potential to change how government acquires or creates IT innovation itself. The main principle is *open by default*: software is developed and delivered once – either with the help of contractors who agree to use agile development methods or built inhouse. The source code is then shared on social coding sites such as Github so that other government agencies don't need to replicate the efforts and a continuous improvement process contributes to the de-siloing of government (Dunleavy & Margetts, 2015; Mergel, 2015).

The second principle is the need for an *agile leadership* approach. While scholars and practitioners might be dismissing the importance of leadership and management in the design implementation of innovation, in risk-averse government environments no action is taken without explicit top-down confirmation. An agile leadership approach (Ryan & Ali, 2016) helps to change deep seated attitudes, values, and habits of the hierarchical bureaucracy. Leadership works toward cultural changes that allow for open collaboration between contractors and internal development teams and changes the perception that the only safe approach to deliver is a complete plan. Agile leaders are responsible for guiding a team to success even in situations where they are inexperienced in agile methods and introduces the culture of prototyping and experimentation using shortened timelines that help to deliver results faster. They also incorporate the possibilities to explore alternatives and allow their teams to fail. The difference to risk-averse failing approaches is that failure in an experimental design approach occurs faster and errors can be eliminated faster – not a year or two into the contract delivery timeline. This procedural innovation creates a cultural change from contract management to participatory and collaborative culture. Agile leaders also have the ability to create an environment and the support for motivated contributors to trust the process and motivate them to participate (Fowler & Highsmith, 2001).

The third design principle includes *alternative contracting and outsourcing approaches* (GAO, 2013; USDS, 2015, 2016). Governments are in the habit of 'big designs up-front.' This is especially true in the U.S. federal government that is driven by budget incentives as well as existing rules and regulations. Firm fixed-price contracts are considered as the common contract type. Full specifications have to be agreed upon upfront, even though most contract managers have little knowledge of IT specifications and actual users of the output are not included in the early design and specification phases. Contractors then need to deliver on the once agreed contract or suggest follow-up contracts. The benefits of a comprehensive agile innovation management approach is that clients, end users, contract managers, and contractors are included in the contract specification phase and leave enough room for adjustments as they go through the implementation and delivery phases. With continuous iterations, small failures are quickly corrected and will lead to successful outcomes earlier (Dullemond, van Gameren, & van Solingen, 2009). The result is that government can create teams of intrapreneurs or entrepreneurs (Schumpeter, 1949) who see government as a start-up opportunity by stepping away from the standard operating procedures of IT innovation and acquisition.

6. Future research on agile innovation management in government

Current research on agile innovation management in government is lacking in the information management literature. Most research that is contributed resides in the computer science field and focuses on scrum

and agile methodologies, however there is limited research that pays attention to the actual implementation challenges, innovative outcomes, or transformation of digital services in government. The following open research questions might help readers focus their own efforts to contribute to the discipline and gain an understanding how governments can overcome their IT legacy problems, move away from stale and risk-averse acquisition routines that might lead to delays in delivery and push contractors to deliver over budget.

Innovative agile acquisition practices: innovative acquisition practices are a learning experience for both – IT contract managers in government as well as IT industry contractors. A recent congressional hearing allowed lobbyist from the software development industry to state their concerns, that some contractors might be left behind who are not willing to offer agile methodologies or who are not willing to reveal their source code for proprietary reasons (Committee on Oversight & Government Reform, 2016). It is unclear at this point to what degree the contracting industry is impacted by changing requirements of government. Additional research is needed to understand a) how agile methodologies are changing the landscape of the IT software industry and specifically vendors who have not offered agile methodologies so far, and b) do these agile blanket purchasing agreements lead to cost savings, more effective and efficient IT contracting practices, and are they truly contributing to increased user- and client-centric outcomes?

The principle '*open by default*' ties into the acquisition procedures that force contractors to make their code open source, as well as the sharing of software code (Crowston, Wei, Howison, & Wiggins, 2012). Currently, in the U.S. case, the principles are followed, however there is no official policy. A draft policy document is circulating with the request for comments and the intention of the White House was publicly stated (Scott, 2016). So far there is little empirical evidence how open source development can lead to sharing of resources or the reduction of double work in government. Just because the source code is open does not necessarily mean that agencies also see the value in it (Raymonds, 1999). Does sharing of software code as seen for example on Github (Mergel, 2015) truly lead to increased collaboration among agencies contributing to the improvement of each others' code, to the actual reuse of already existing code, and subsequently to reduced spending of tax payer money? In addition, do existing open source policies lead to the expected change in acquisition procedures? Recently, the U.S. federal government even unpublished its source code for the problematic HealthCare.gov website (Jeffreis, 2013). Do vendors – and governments – comply with the policies? As an example, a recent Congressional Hearing of the Oversight Committee in the U.S. has shown that the software industry is against open source development (Committee on Oversight & Government Reform, 2016). How can government incentivize open sharing of source code instead of reinventing the wheel with every request for proposals, signed contract or grant? Some insights might be gleaned from other types of sharing services outside the software development problem.

Another important effort focuses on *scaling up practices and replicating lessons learned* from the digital swat teams in the White House across the U.S. federal government. As Boehm and Turner (2005:30) point out: "on small, stand-alone projects agile practices are less burdensome and more in tune with the software industry's increasing need for rapid development and coping with continuous change," however, they also observe the "difficulty scaling up and integrating them into traditional, top-down systems". Especially in government, where organizations are expending 80% of their budget on maintaining legacy systems, it is unclear how agile innovation management practices can be scaled up from the project level to transforming overall digital approaches for each agency and government as a whole. It is important to understand how open by default practices can be adopted and implemented in the current environment that is still extremely risk averse. While there is evidence that projects that adhere to agile development concepts were nearly twice as likely to deliver on time than those that used the traditional "waterfall" development technique, it is still a

show-and-tell effort to convince agency leadership and especially IT contracting managers to buy into the concepts. Future research can focus on diffusion of agile practices, but also on comparative studies to highlight the differences between a traditional contracting approach and an agile innovation management approach.

The main challenge for government is the *cultural change* that needs to go hand-in-hand with the procedural changes. The current notion of 'small vs. big government' has shifted budget categories toward contracting out and led to a reduction in IT talent that is located in government. Instead, contractors' and vendors' knowledge is moved from the outside of the organization into government on a temporary basis, but is again removed after a contract ends. How can government attract IT talent from the private sector to serve? This is an important question given the gap in salaries paid by the private sector and the limits to pay out bonuses or create flexible salary schemes. Once recruited, "inculturating" especially Silicon Valley-type software engineers for whom agile methods are the norm into a risk-averse contracting-out environment is as Christy (2016) states a "big mind shift" for government employees. A top-down commitment to openness and transparency by design is necessary to shift the internal paradigm, but also many of the barriers and challenges of the current work environment have to be adapted to allow for changes toward the start-up culture that software engineers value in their industries. Here it is important to understand new forms of prosocial public service motivation, that goes beyond the intrinsic motivations to do good or contribute to the public sector (Grant, 2007). Self-determination theories contradict the traditional command-and-control environments in which government bureaucracies are designed (Gagné & Deci, 2005). These theories might go beyond the motivation to contribute to a public good as most of the current literature on open source programmers suggests (Blasc, Jung, & Lakhani, 2016; Budhathoki & Haythornthwaite, 2013). It is therefore important to gain a deeper understanding how government can attract and more importantly also retain IT talent that has other attractive options outside the restricted environment of government. What kind of policy and management changes are necessary to accomplish these goals?

Agile public leadership is needed to initiate process changes from the top, however it requires middle management support to empower and showcase change that is possible when using an agile innovation management approach (Riby & Sutherland, 2016). Leadership is needed when it comes to providing adequate coverage top down. This will lighten the risk-aversion and will help managers to push through barriers that have to do more with the practiced and proven approaches, but are usually not even in the law or in any kind of other bureaucratic norms (Ryan & Ali, 2016). It is necessary to change collaborative innovation creation and sharing culture in government. As Shaikh (2016) showed in a recent study, the adoption of open source software as the standard approach in government is challenging: while it creates possibilities of community development, sharing, and reuse in government, there are also many barriers and clear policy changes are needed to help government organizations to adopt an open source strategy. One example, is the U.S. federal government's Federal Source Code draft policy document (Scott, 2016). This policy draft suggests that "new software developed specifically for or by the Federal Government to be made available for sharing and re-use across federal agencies." However, there is little evidence so far, that leadership across the federal government is in support and more research is needed to understand how top-down policies with an agile-minded leadership can provide adequate support for broad scale change initiatives.

Lastly, Presidential initiatives are prone to disappear when an administration transitions to its successor. National priorities might shift, budget cuts might move temporary financial support from one priority to a new project, and IT talent recruited by the former administration might return to the private sector. While there are efforts to smoothly transition innovative initiatives from one administration to another, projects that are not institutionalized or are seen as pet projects of a

former President might not survive the transition. Future research needs to address how the electoral cycle and shifting national priorities affect the adoption of agile methodologies, how commitment to innovation management persists, and whether risk-taking with government technology decisions led to sustainable changes, such as the enactment of acquisition policies and innovative HR policies.

Overall, agile innovation management is an approach that many different countries are attempting – with some governments in advanced stages, others in early experimentation stage. Most of the topics listed above are either not researched or in isolation. It is therefore important for e-government and digital innovation researchers to conduct country-level studies, project studies, as well as human resources studies to increase our understanding agile innovation management.

References

- Agile Manifesto (2001). Manifesto for agile software development. (Accessed 05/10/2016) <http://agilemanifesto.org/>.
- Amazon.com. (2016). Diego Piacentini: He will take a two-year leave to head the Italian Prime Minister's digital technology office.
- Anthopoulos, L., Reddick, C. G., Giannakidou, I., & Mavridis, N. (2016). Why e-government projects fail? An analysis of the healthcare.gov website. *Government Information Quarterly*, 33(1), 161–173.
- Balter, B. J. (2011). Toward a more agile government: The case for rebooting federal IT procurement. *Public Contract Law Journal*, 41(1), 151–171.
- Bertot, J., Estevez, E., & Janowski, T. (2016). Digital public service innovation: Framework proposal. *ICEGOV2016, Montevideo, Uruguay, March 1–3, 2016*.
- Blasc, A., Jung, O. S., & Lakhani, K. R. (2016). *Motivating effort in contributing to public goods inside organizations: Field experimental evidence*.
- Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software September/October:30–39*.
- Broadus, W. (2013). The challenges of being agile in DoD. *Defense AT&A, January–February: 5–9*.
- Budhathoki, N. R., & Haythornthwaite, C. (2013). Motivation for open collaboration crowd and community models and the case of OpenStreetMap. *American Behavioral Scientist*, 57(5), 548–575.
- Christy, A. (2016). Government goes agile. *Stanford Social Innovation Review Spring, 2016*.
- Cockburn, A. (2002). Learning from agile software development – Part one. *The Journal of Defense Software Engineering*, 15(10), 10–14.
- Collier, D. (2011). Understanding process tracing. *PS: Political Science and Politics*, 44(4), 823–830.
- Committee on Oversight and Government Reform (2016). 18F and U.S. digital services oversight. *Congressional hearing*.
- Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)*, 44(2) (n.p).
- Department of Defense Appropriations Act (2010). *H.R.3326 111th Congress (2009)*.
- Drever, E. (1995). *Using Semi-Structured Interviews in Small-Scale Research: A Teacher's Guide*. Edinburgh: Scottish Council for Research in Education.
- Dullemond, K., van Gasteren, B., & van Solingen, R. (2009). How technological support can enable advantages of agile software development in a GSE setting. *Fourth IEEE International Conference on Global Software Engineering*.
- Dunleavy, P., & Margetts, H. (2015). Design principles for essentially digital governance. *11th Annual Meeting of the American Political Science Association, San Francisco, 3–6 September 2015*.
- Dunleavy, P., Margetts, H., Bastow, S., & Tinkler, J. (2005). New public management is dead – Long live digital-era governance. *Journal of Public Administration Review and Theory*, 16, 467–494.
- Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. Retrieved from <http://agilemanifesto.org/>.
- Fulgham, C., Johnson, J., Crandall, M., Jackson, L., & Burrows, N. (2011). The FBI gets agile. *ITProfessional*, 13(5), 57–59.
- Gagné, M., & Deci, E. L. (2005). Self-determination theory and work motivation. *Journal of Organizational Behavior*, 26(4), 331–362.
- GAO (2012). Software development: Effective practices and federal challenges in applying agile methods. *Report to the Subcommittee on Federal Financial Management, Government Information, Federal Services, and International Security, Committee on Homeland Security and Governmental Affairs United States Senate*. Washington, DC: Government Accountability Office.
- GAO (2013). Leveraging best practices to help ensure successful major acquisitions. In Government Accountability Office (Ed.), *Information technology*. Washington, DC: GAO.
- GAO (2014). *HEALTHCARE.GOV: Contract planning and oversight practices were ineffective given the challenges and risks*.
- GAO (2015). Information technology: Additional actions and oversight urgently needed to reduce waste and improve performance in acquisitions and operations. In Government Accountability Office (Ed.), *GAO-15-675T*.
- Goldman, S. L., & Preiss, K. (1994). *Agile competitors and virtual organizations: Strategies for enriching the customer*. San Francisco, CA: Wiley.
- Grant, A. M. (2007). Relational job design and the motivation to make a prosocial difference. *Academy of Management Review*, 32(2), 393–417.

- Janssen, M., & van der Voort, H. (2016). Adaptive governance: Towards a stable, accountable and responsive government. *Government Information Quarterly*, 33(1), 1–5.
- Jeffreis, A. (2013). "Why the government unpublished the source code for Healthcare.gov." TheVerge.com. <http://www.theverge.com/2013/10/18/4852720/why-the-government-unpublished-the-source-code-for-healthcare-gov-github>.
- Lapham, M. A. (2012). DoD agile adoption: Necessary considerations, concerns, and changes. *CROSSTALK - The Journal of Defense Software Engineering* January/February: 31–35.
- Leetaru, K. (2016). System failure: What I've learned as a data scientist in Washington. *Forbes*.
- Margetts, H., & Dunleavy, P. (2013). The second wave of digital-era governance: A quasi-paradigm for government on the web. *Philosophical Transactions of the Royal Society*. <http://dx.doi.org/10.1098/rsta.2012.0382>.
- Martin, R. C. (2003). *Agile software development: Principles, patterns, and practices*, Alan Apt series. Upper Saddle River, NJ: Prentice Hall Pearson Education, Inc.
- Mergel, I. (2015). Open collaboration in the public sector: The case of social coding on github. *Government Information Quarterly*, 32(4), 464–472.
- Office of Management and Budget (2011). *Implementing Executive Order 13571 on streamlining service delivery and improving customer service*.
- Raymonds, E. S. (1999). *The Cathedral and the Bazaar: Musings on Linux and Open Source*. Sebastopol, CA: O'Reilly Media.
- Read, R. L. (2016). Don't ask for permission to use agile. *Just start doing it. NextGov*.
- Riby, D., & Sutherland, J. (2016). Understanding agile management. *Harvard Business Review*.
- Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016a). Embracing agile. *Harvard Business Review*.
- Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016b). The secret history of agile innovation. *Harvard Business Review*.
- Royce, W. (1970). *Managing the development of large software systems*. (IEEE WESCON).
- Ryan, K., & Ali, A. (2016). The new government leader: Mobilizing agile public leadership in disruptive times. In Deloitte University Press (Ed.), *A GovLab report*.
- Schumpeter, J. A. (1949). Economic theory and entrepreneurial history, reprinted from change and the entrepreneur. In R. V. (Ed.), *Essays on entrepreneurs, innovations, business cycles, and the evolution of capitalism*.
- Scott, T. (2016). "Title." The White House: What is Happening. (03/10/2016) <https://www.whitehouse.gov/blog/2016/03/09/leveraging-american-ingenuity-through-reusable-and-open-source-software>
- Shaikh, M. (2016). Negotiating open source software adoption in the UK public sector. *Government Information Quarterly*, 33, 115–132.
- Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Thousand Oaks, CA: Sage Publications.
- Surdu, J., & Parsons, D. J. (2006). Army simulation program balances agile and traditional methods with success. *CROSSTALK - The Journal of Defense Software Engineering* April: 4–8.
- Sutherland, J. (2012). "Title." Scrum Blog. (07/18/2012) <https://www.scruminc.com/dod-goes-agile/>
- Takeuchi, H., & Nonaka, I. (1986). The new. New product development game. *Harvard Business Review*, 64(1), 137–146.
- The White House (2011). Executive order 13571–streamlining service delivery and improving customer service. *Executive order 13571*.
- The White House. 2012a. Contracting guidance to support modular development. In Procurement, edited by Office of Management and Budget. (Washington, DC).
- The White House (2012b). *Digital government: Building a 21st century platform to better serve the American people*.
- The White House (2012c). *Presidential memorandum – Building a 21st century digital government*.
- The White House (2015). *Executive order – Presidential Innovation Fellows Program*.
- Torgovnick, M. K. (2016). What happens when you disrupt the White House: Haley Van Dyck speaks at TED2016. *Live from TED2016*.
- U.S. DoD (2015). *Secretary of defense speech: Drell lecture: "Rewiring the Pentagon: Charing a new path on innovation and cybersecurity"*.
- UK.gov. (2016). *Government digital service*.
- USDS (2015). *U.S. digital services playbook*. (Washington, DC.).
- USDS (2016). *The TechFAR handbook*.

Dr. Ines Mergel is full professor of public administration at the University of Konstanz, Germany, where she teaches classes on government innovation management. Professor Mergel received a Doctor of Business Administration and spent six years as a postdoctoral fellow at Harvard's Kennedy School of Government. From 2008 to 2016, she taught in the Master of Public Administration Program at Syracuse University's Maxwell School of Citizenship and Public Affairs. Her research focuses on technology management innovations mostly in the U.S. federal government.