Advancing the Agile Software Process: The case of modernizing the Army Community Service's information technology infrastructure

Jessica A. Pater
Information and Communications Lab
Georgia Tech Research Institute
Atlanta, GA USA
Jessica.Pater@gtri.gatech.edu

Samantha Lie-Tjauw
nformation and Communications Lab
Georgia Tech Research Institute
Atlanta, GA USA
Samantha.Lie-Tjauw@gtri.gatech.edu

Margarita Gonzalez
nformation and Communications Lab
Georgia Tech Research Institute
Atlanta, GA USA
Margarita.Gonzalez@gtri.gatech.edu

Moon Kim
nformation and Communications Lab
Georgia Tech Research Institute
Atlanta, GA USA
Moon.Kim@gtri.gatech.edu

Sheila Isbell
nformation and Communications Lab
Georgia Tech Research Institute
Atlanta, GA USA
Sheila.Isbell@gtri.gatech.edu

David Severson
Army Community Services Division
IMCOM G-9, Family & MWR Programs
San Antonio, TX USA
david.b.severson.civ@mail.mil

***Abstract*— There is a widening gap between the promises of big data, interoperability, and service-oriented systems and what most governmental agencies currently receive in return from their information systems. Such disparity is apparent in the domain of health and human services, as government organizations are grappling with the challenge of streamlining their business processes and delivery of care to adhere to a person-centered approach that relies on interoperable systems and meaningful data exchange. Since 2014, the Georgia Tech Research Institute has supported the United States Department of the Army in this endeavor by helping to modernize a suite of information systems used by the Army to provide family-centric social services, in support of family readiness and inclusive of child wellbeing, healthy marriages and parenting, and the treatment and prevention of domestic violence and child abuse. In this paper, the authors discuss how agile software development methodology on its own is not conducive to modernization efforts where legacy systems, legacy data, and legacy human processes cannot be replaced entirely. Instead, the authors propose an adapted model that combines agile development with enhanced user-centered design techniques as a socio-technical approach to software modernization. We will describe our process, reflect on lessons learned, and discuss select future implications associated with this type of work.**

Keywords—social technical systems, agile software development, software modernization, data integration, business processes, reporting, qualitative methods, computer supported cooperative work, enterprise, service delivery, modularity

I. INTRODUCTION

Software development methodologies continue to evolve to accommodate the fast-changing information technology (IT) needs of businesses and organizations. Increasingly, agile software development and the broader DevOps approach are becoming common practice among software development and IT professionals. The underlying motivation for this shift is that the agile approach provides an integrated framework that makes possible the flexibility to identify problems sooner rather than later, enables collaboration with end-users and fosters a faster feedback cycle. At the heart of this evolution is the need for greater collaboration, more seamless integration of disciplines, and effective communication across an enterprise. This imperative is not just applicable to the development of enterprise-wide IT systems, but also in the design, specifically as it pertains to the encapsulation of enterprise-wide business processes that are usually not silo-ed even if the IT systems intended to support them are. Legacy IT systems typify data captured and stored without regard to how other business processes or functions may use or update this data. Often, user workflows morph in order to accommodate the IT shortfalls, evidenced by the serial, manual updates of the multiple systems that they access to support any given business process. Unfortunately, the labor-intensive, error-prone processes contribute to organizational inefficiencies and employee overtasking.

The widening availability of cloud computing technologies and the business pressure to do more with less are driving organizations to take on large IT modernization efforts. However, it is not financially feasible nor business practical to carry-out enormous "rip and replace" initiatives. Leveraging an agile framework enables targeted, cross-functional collaboration throughout the development cycle necessary to transform a monolithic, "data-centered" legacy system into an IT system that better supports the enterprise's broader socio-technical context; ultimately, eliminating data redundancy and business process fragmentation. To maximize the impact of a modernization effort, careful attention should be given to aligning diverse user IT needs with broader organizational

data needs—which is a non-trivial matter. This is further complicated when the impact of nascent policy and business mandates on multi-tier, cross-functional business processes are ambiguous and the IT/data requirements necessary to support the modified business process are not yet ascertainable.

Few organizations have the luxury of completely re-engineering their business processes from end to end in support of a large modernization effort. This is especially the case in the context of health and human services where the legacy system limitation is exacerbated by burdening caseloads, alarming staff turnover, and a decrease in resources. In this environment, an ill-planned modernization effort could cause a devastating disruption to the delivery of clinical and non-clinical services. However, by modularizing the modernization effort, technical teams can support targeted business process improvements that serve as the foundational blocks for evolving the technology landscape of health and human services.

In this paper, we will describe the efforts to adapt an agile development model with enhanced user-centered design techniques in order to modernize a suite of complex health and human service information systems for a closed community. We will describe our process, reflect on lessons learned, and discuss select future implications associated with this type of work.

II. Background

According to the report by the U.S. Government Accountability Office, the federal government spent over 75% of its information and technology funding on operations and maintenance (O&M) in 2015 [15]. Moreover, from 2010 to 2017 spending on development and modernization efforts dropped $7.3 billion while O&M spending continued to rise. Although the IT industry is anticipating trends like cloud migration, data analytics, and machine learning in government agencies [15, 20], the benefits from these advancements will not be attainable without addressing the need to modernize their legacy systems. As such, there is a real need and focus across all levels of government to modernize their IT infrastructure [4, 9].

As noted earlier, there are increasing mandates in health and human services to cut costs and streamline services. Most of these mandates are tied to the larger goal of delivering a person-centric approach to healthcare and human services as specified by statutes like the Affordable Care Act [21]. The IT needs of health and human services present an interesting case study in how the policy goals point towards coordination and interoperability while their state of technologies is still mired in legacy infrastructure that are silo-ed and do not support the workflows necessary for healthcare delivery in the 21st century.

As of 2017, the United States' Department of Health and Human Services (DHS) constitutes a significant amount of total government spending, with a current budget of $82.2 billion in discretionary funding. Critical funding priorities for the 2017 budget include precision medicine initiatives, building an evidence base to drive systemic health care improvement, and promoting information sharing [16]. Given the push to drive down costs while improving quality of care, the health sector is ripe for innovation. One such innovation is the concept of "lean healthcare", adopted from lean principles applied to the auto industry, which necessitates timely and comprehensive information sharing to eliminate "waste to improve the flow of patients, information or goods" [18]. Many innovations in the health care sector include an aspect of improved information sharing which is established through a supported information infrastructure. Historically, the lack of integrated information infrastructures spanning the diverse array of health and human services constrained an institution's ability to engage both service providers and clients/patients with the timely and detailed information needed to provide quality and patient-centered care [5]. Rycroft-Malone and Seers outlined the importance of a wide breadth of contributors (client, patient, social worker, clinician, etc.) in order to deliver true client/patient-centered care and ultimately support evidence-based care [17]. What many authors including Rycroft-Malone and Seers acknowledge but do not address in detail in the literature is that technology needs to function as the tool for the timely and accurate capture and analysis of these critical data.

Traditionally, information systems in the healthcare and human services domains have focused primarily on information collection and sharing. With the shift towards personalized health care, systems are transforming to provide more tailored decision support and can be positively exploited for opportunities related to big data collection for population-level evidence-based care. However, with new promises also come new challenges. Barriers to widespread adoption and standardization of information systems within healthcare still exist. Barriers include the ability to translate results from health data analytics into impactful actions that influence service providers and the desire for autonomy among service providers in regards to how they provide care [8].Using development approaches like Agile software development can offer tools to address how these barriers are identified and addressed within the software and the policies that envelope the deployment and utilization of the software.

Several mainstream software development models include waterfall, prototyping, spiral development and rapid application development. Agile development is considered common practice in most modern development efforts [10] and is a reaction to these traditional ways of developing software [2]. This model has four key concepts – individuals and interactions, working software, customer collaboration, and responding to change [2]. While there are still spaces where other models are sufficient, the dynamic and rapidly changing world increasingly requires processes that are agile and flexible [10]. Since the introduction of Agile development processes in 2001, practitioners have applied this methodology to a variety of disciplines. User centered design [3], business intelligence [13], data analytics [14] and security [12] are examples of this integration. The Agile development cycle is comprised of six key phases: requirements, plan, design, develop, release, and track and monitor [6]. Through these integration efforts, these phases been transformed and expanded. Larson and Chang adapted this process to align with business intelligence and data science frameworks. Figure 1 shows how they transformed the standard Agile development process into the Data Science domain.

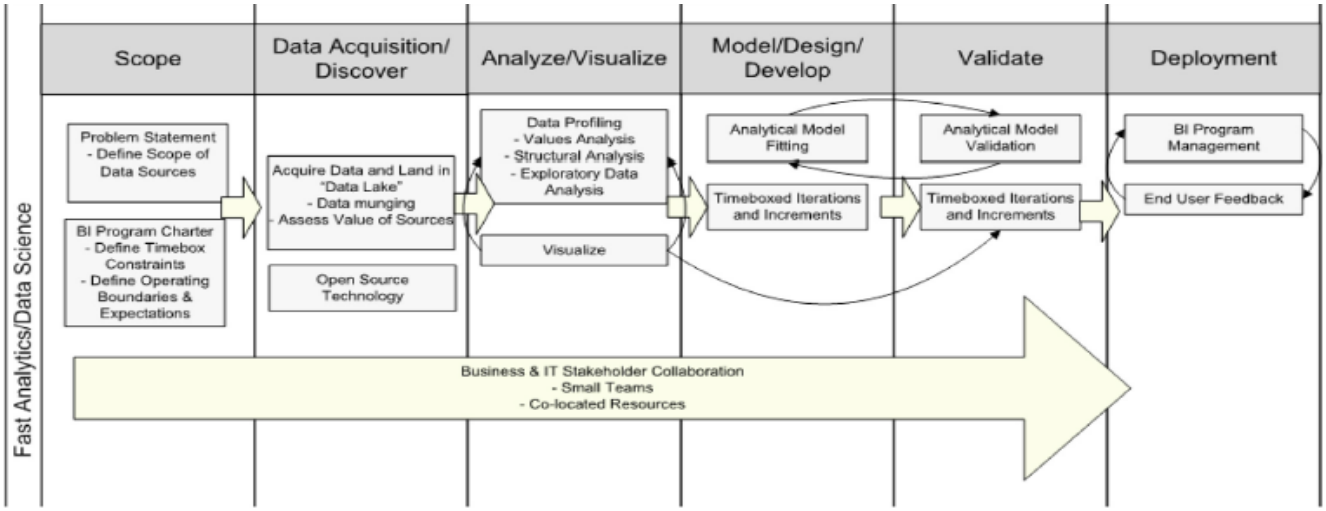


Figure 1. Larson and Chang Agile Development Cycle application to Data Science

III. METHODOLOGY

In order to address the modernization of enterprise-wide IT systems within the health and human services domain, we first sought to establish the frameworks that guided the team's organization of the problem space. We chose to view this domain through the lens of an adapted model that combines agile development with enhanced user-centered design techniques as a socio-technical approach to software modernization.

A. STS

Systems focused in the health and human services domain are inherently complex in nature and are exemplars of sociotechnical systems (STS). Sociotechnical Systems is an organizational development approach to complex design that puts the interaction between people and technology at the forefront [1]. When modernizing a set of complex systems where an end-user is needed to ensure the software meets work-practice needs, STS is the preferred conceptual framework because of its vast body of work underlying employee involvement and work design applications [1]. This close collaboration often means that as technical systems evolve, so do the social systems (or policies) that guide their use [7]. In this work, we chose to use STS as a guiding lens for our adaptation of the Agile framework.

B. User-Centered Design Tools

A critical component of adopting and executing the STS conceptual framework within the Agile development cycle involved selecting a variety of user-centered design strategies. We leveraged a combination of user-centered design techniques along with elements from participatory design, empathic design, and design thinking to guide the data collection methods. The common thread between each of these frameworks was maintaining the user at the center of the process and regarding the user as the expert [11]**.** Thereby, the data collection methods utilized provided continuous user-driven input into each phase of the Agile development cycle [3]**.** The mix of qualitative and quantitative data collection tools provided us with a deep understanding of the users and their socio-technical context driving their data needs often buried in their sophisticated work-arounds and technology "pain-points." Additionally, our mixed-methods data collection tools enabled us to tap into users' tacit knowledge, encouraged user agency and invited them to co-create the solution. Depending on the tool and the phase within the project, implementing the method either required the team to delve into the world of the users (empathic design and design thinking) or invite the users into the world of research and development (user-centered design and participatory design) [3, 19]**.** The data collection tools included: key informant interviews, in-depth interviews, surveys, design feedback sessions, observations, and focus groups.

C. Adapting the Agile Framework

As previously mentioned, the affordances and popularity of the Agile framework has led to the adoption and adaptation within specific domains. For the modernization of software systems that are data-driven, we feel that this is a more relevant model to use than the traditional standard Agile framework previously described. In the traditional models, direct user engagement is often heavily focused to the cycles at the beginning and end of development [3]. The Larson and Chang model focused on Data Science breaks with this model and creates greater possibilities along the software development continuum [13]. In the preparation for modernizing a suite of legacy software systems we quickly realized that we would need to further adapt the Data Science model to

ensure that diverse workflows and user constraints (in-situ) were prioritized during the modernization process. Table 1 highlights how we merged user-centered design practices into each step of the process in addition to the translational activities/outputs that were derived from these practices.

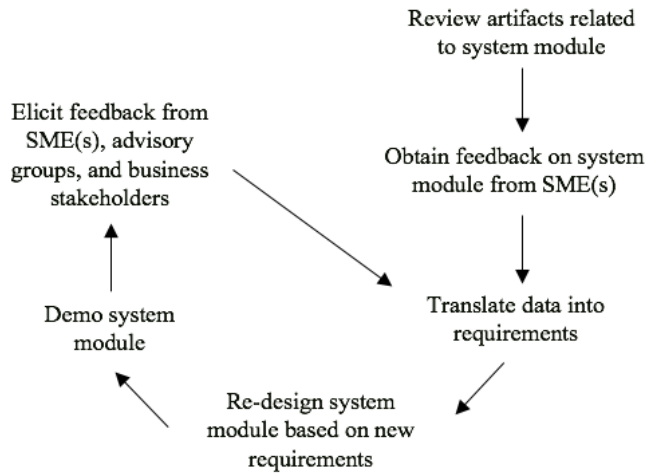
	Scope	Data Acquisition	Analyze/Visualize	Model/Develop/ Design	Validate	Deployment
User Centered Design Approaches	Policy Review Interviews Contextual Inquiry Contractual Review Review of legacy software tickets Review of unstructured legacy system user feedback Review of legacy system user documentation	Artifact collection and analysis Administer surveys Document informal policies and procedures Uncover tacit business rules Expose complexity/diversity of embedded data cultures and data literacy levels across the enterprise	Validate Service Delivery Process Maps Deploy in-situ technology assessments Create low-fidelity prototype of proposed design and evaluate user feedback	Validate requirement matrix with SMEs and diverse business owners Conduct user testing on minimally viable product (MVP) of specific system modules Form program-specific advisory groups to support the co-creation process	Conduct user testing on in-progress modernized systems Hold Stakeholder engagements to review finalized modules and the system as a whole	Finalize training materials with SME / advisory groups Collect deployment feedback
Translational Activities	Validation of legacy system benefits Root cause analysis of pain points	Develop Service Delivery Process Maps Translate user pain points Operationalize survey data Develop user profiles & stories Develop technical and policy opportunities matrix based on pain points and benefits of current system Create user testing materials for structured feedback			Develop training materials (Guides, videos, training, etc.) Develop deployment communication packages Develop and validate roll-out strategy (single vs phased) Create initial deployment assessments	Analyze deployment feedback Translate feedback into future functional requirements

Table 1. Adaption of Larson & Chang Fast Analytics/Data Science Agile Model for Software Modernization

IV. Case STudy – Army Community Services (ACS) and the Army Family Web Portal (AFWP)

Since 1965, Army Community Service (ACS) programs have held a special role in contributing to the stability and resiliency of the Army through the provision of needed social services, such as parental education, treatment and prevention of domestic violence, financial education, and support for relocating soldiers and families. ACS programs aim to provide comprehensive, coordinated, and responsive services that support the readiness of soldiers and their families. These services are broken into three key areas: family advocacy, volunteer, and readiness. These programs use a suite of individual information systems that ACS program staff use to manage the data and services provided to Army Families (See Table 2). In total, there are seven information systems.

To advance this mission, Army Installation Management Command (IMCOM) G-9 partnered with the Georgia Tech Research Institute in 2014 to modernize this suite of information systems to support enhanced client management and service delivery. This work is currently entering its fourth year. Over the course of three years, this suite of seven systems and the new AFWP core platform have been in iterative and intense development. The suite of systems will all publically deploy throughout this operating period.

The key driver of this effort was to move these information systems to a cloud-based architecture. Once in the cloud these individual systems were then integrated into a new singular platform, the Army Family Web Portal (AFWP). Figure 3 highlights the platform design of the AFWP. This unification will allow for shared user profile information, single sign-on for the suite of information systems, and enhanced reporting capabilities.

	Programs	Systems
Readiness	•Relocation •Mobilization and Deployment •Family Readiness Group •Financial Readiness •Employment Readiness •Lending Closet	•ACS Staff •CMS •OLMS •vFRG •AOS
Family Advocacy	•Exceptional Family Member Program •New Parent Support Program	•ACS Staff •CMS •eEFMP
Volunteer	•Army Family Team Building •Army Family Action Plan •Army Volunteer Corps	•ACS Staff •IMS •OLMS •VMIS

Table 2. ACS programs and the information systems used to support them.

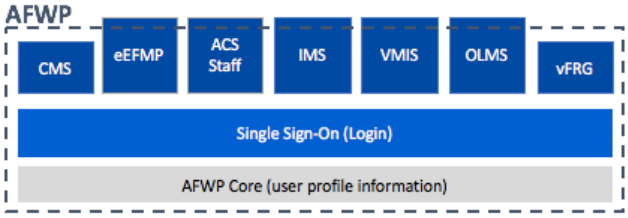


Figure 3. AFWP Platform design

A. Population

This case study is situated within the United States Army. Throughout this process, the research team had extensive access to U.S. Army staff, contractors, enlisted Service Members, and Families. In total, we engaged over 420 participants at 49 installations across the world through a combination of virtual and in-person means. To gain access to these individuals, we worked with IMCOM headquarters to identify the ACS staff and program subject matter experts and then employed snowball sampling techniques to uncover who they recommended would be the most knowledgeable or helpful with the task-at-hand. We ensured that the installations that we engaged spanned in size (small, mid-sized, large) and focus (readiness, sustainment, training). It should be noted that this is a closed system – for the most part, all engagements are closed off from the outside environment and all interactions and knowledge is transmitted within the system.

B. Data Collection

Based on the outline of the user-centered design tools from the previous section, we employed a wide range of tools to capture necessary mixed-method data. Table 3 highlights the range of tools used and how many times they were used during the research process. Table 1 describes the phases of research when these tools were deployed. The data collected through these tools were critical to streamlined design and modernization of the legacy systems. Additionally, the mixed-method data informed the team's

UCD Method	Number of Tools Used	Explanation of Activity(s)
Key Informant Interviews	16	Semi-structured interviews to develop initial understanding of socio-technical context
In-Depth Interviews	83	On-the-ground interviews to elicit insight into business processes and supporting workflows
Surveys	15	Digital qualitative surveys targeted based on/to validate interview data
Design Feedback Sessions	28	In-person sessions to review and discuss initial system modular development
Observations	15	Direct observation to assess business processes within context
Focus Groups	25	Users grouped strategically gathered to validate findings, discuss workflow improvements, and provide feedback on proposed module designs

Table 3. UCD Tools used in case study

recommendations for updating/automating, or developing workflows in support of business process improvements to support the legacy system modernization effort. Internally, we divided our core team of human-centered design researchers into the individual system development teams. We met multiple times a week to share updates and best practices as well as work collaboratively on the integration of the research insights in the technical design, development and deployment of each system. By tackling the modernization effort in a modular fashion, this multi-disciplinary team environments allowed for cross-pollination of system architecture insights that accelerated the team's ability to diagnose and address pain points in other modules.

C. Lessons Learned from the case study

The team learned several key lessons from this modernization effort. First, unlike other systems that we build from the ground up, these systems had been deployed for well over a decade. There was a wealth of institutional knowledge within the program offices, the sub-contractors that maintained the system, and the end users that use these systems daily. Breaking through the status quo, establishing rapport, and understanding the complexity and intricate nature would have taken too long if it were all to take place up-front – a common critique of user centered design processes [3]. Thus, the interdisciplinary team struck a compromise between the two processes, creating a hybrid approach that allowed for greater flexibility as new data came in from the field. The social scientists in the field collecting and analyzing the data took on new tasks of rapidly translating findings into software requirements. This was made possible by the modular and targeted approach taken by the development and research teams. To ensure that multiple iterations could take place within each step of the model, the developers and the designers worked in tandem to ensure that all assessments were highly structured and focused to a specific module, enabling a quicker translation of needs from the user base into the development stack.

Another key takeaway was broadening our stakeholder engagement beyond that of traditional software development processes. Typically, business owners generate a stack of requirements for the software to address identified needs. Within a user-centered design approach, a designer will work with key subject matter experts to identify further tailored requirements for the ecosystem in which the software will be ultimately used. Working with our business owners, we purposefully and routinely engaged a wider set of people in the requirement generation process. We assembled teams of end users, people impacted by the outcomes of the systems, managers of end users, policy makers within the ACS system, business owners, and technical staff from within the ACS system to give a more holistic and nuanced grounding for the new software modernization efforts. Through this approach, we uncovered obscured inconsistences between policy, legacy system configuration, and on-the-ground utilization. The modernization efforts of the new systems are addressing these issues, and hopefully will alleviate some of the burden that was inherent in the previous configuration.

V. DISCUSSION

A vast majority of data currently collected by information systems within social services provided to military families as well as those provided to civilians at large can be categorized as: transactional, personal health information (PHI), incident, situational awareness, administrative, and training utilization. Social determinant data is missing from these data categories. Taking the opportunity to capture and share information regarding education and job history, social support, attitudes and practices, religious beliefs, exposure to violence, literacy, and socioeconomic conditions can allow to not only gain a window into understanding the origin of a client/patient's specific issue, but how to best address that issue within an individual's current reality. Furthermore, the importance of such information aligns with the World Health Organization's Health People 2020 initiative.

Through the inclusion of such data, both client/patient and service provider gain a more comprehensive and nuanced picture of the individual. All the former types of data exist within a context and that context is often driven by social determinants. Understanding the environment in which an individual lives, works, and plays allows for a deeper understanding of an individual's total wellness (physical, emotional, mental, and spiritual health). In turn, this deeper knowing provides an opportunity to personalize services through a client-centered mindset. However, the addition of these data can bring challenges along with value. Challenges include the accurate capture of social determinants data due to the nature of it being self-reported or relying on its capture within other trusted sources. Additionally, these data are more powerful when coupled with the existing transactional, PHI, administrative data etc, but may require currently disparate information systems to share data in timely and complete fashion. Lastly, but perhaps most importantly is the lack of training to service providers on how to effectively act upon this information even if it is available.

A point to revisit—If legacy IT systems do not adequately support an organization's business processes this will create the inevitable need for users to compensate for the IT shortfalls. This is demonstrated by creative work-arounds users improvise in order to capture necessary data within the constraints of their technology. Alternatively, users capture data outside of these systems on shared drives, local desktops, and non-digital artifacts to name a few of the more common practices. This is an example of an unintended impact on business processes that ultimately further fragments the enterprise's data, compromising its integrity and reliability. That said, as part of an enterprise IT modernization effort, thoughtful planning should be given to the inclusion of new data beyond the traditional transactional data typical of monolithic systems. Data types and elements should reflect the workflows that constitute broader business processes. In order to mitigate the burden of data collection, thought should be given enabling distributed rather

than centralized data collection. To that end, leveraging robust access management principles would make possible the inclusion of new system users that can share the responsibility of data entry while mitigating security and user access concerns. In addition to collecting data that more holistically supports any given business process, this same data can be mined to make better informed decisions.

VI. Conclusion

In the near future, there are scores of legacy information technology systems that are maturing and in need of modernization. Large enterprise sociotechnical systems are prevalent in the health and human services domain and will continue to grow in other domains as machine learning, AI, and data analytics becomes more accessible and easy to integrate into data driven systems. As these systems expand it is important that the community find more meaningful ways to give end users and people generating the data more advocacy and presence within the design and execution of these systems. As we build systems that are more reflective of current end users and policies, we also need to be mindful of designing extensibility into these systems that will allow them to grow and evolve with the programs they support.

Acknowledgment

We would like to acknowledge our ARFAM team at GTRI for their support and dedication to the use of these methods. Additionally, we would like to acknowledge the Army ACS Programs and the countless staff, service members, and families that have contributed to these modernization efforts.

References

[1] Appelbaum, S.H. 1997. Socio-technical systems theory: an intervention strategy for organizational development. *Management Decision*. 35, 6 (1997), 452–463.

[2] Beck, K. et al. 2001. *Manifesto for Agile Software Development*.

[3] Chamberlain, S., Sharp, H. and Maiden, N. 2006. Towards a Framework for Integrating Agile Development and User-Centred Design. *LNCS*. 4044, (2006), 143–153.

[4] Charette, R.N. 2016. Dragging Government Legacy Systems Out of the Shadows. *Computer, IEEE Journals & Magazines*. 49, 9 (2016), 114–119.

[5] Chassin, M., Galvin, R. and Quality, N.R. on H.C. 1998. The Urgent Need to Improve Health Care QualityInstitute of Medicine National Roundtable on Health Care Quality. *JAMA*. 280, 11 (1998), 1000–1005.

[6] Cohen, D., Lindvall, M. and Costa, P. 2003. Agile Software Development. *DACS SOAR Report*. 11, (2003).

[7] Cooper, R. and Foster, M. 1971. Sociotechnical Systems. *American Psychologist*. 26, (1971), 467–474.

[8] Fichman, R., Kohli, R. and Krishnan, R. 2011. Editorial Overview—The Role of Information Systems in Healthcare: Current Research and Future Trends. *Information Systems Research*. 22, 3 (2011), 419–428.

[9] Fisher, S. 2017. *What to expect: 2017 IT trends in government and the public sector*.

[10] Highsmith, J.A. 2002. What Is Agile Software Development? *The Journal of Defense Software Engineering*. 15, 10 (2002), 4–9.

[11] Improving the User Experience: 2017. *www.usability.gov*.

[12] Keramati, H. and Mirian-Hosseinabadi, S.-H. 2008. Integrating software development security activities with agile methodologies. *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference* (2008), 749–754.

[13] Larson, D. and Chang, V. 2016. A review and future direction of agile, business intelligence, analytics and data science. *International Journal of Information Management*. 36, 5 (2016), 700–710.

[14] Lee, G. and Xia, W. 2010. Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*. 34, 1 (2010), 87–114.

[15] Powner, D.A. 2016. Information Technology: Federal Agencies Need to Address Aging Legacy Systems.

[16] President's HHS FY2017 Budget Factsheet: *https://www.hhs.gov/about/budget/fy2017/budget-factsheet/index.html%0D1.%09https://www.hhs.gov/about/budget/fy2017/budget-factsheet/index.html%0D*. Accessed: 2017-10-07.

[17] Rycroft-Malone, J., Seers, K., Titchen, A., Harvey, G., Kitson, A. and McCormack, B. 2004. What counts as evidence in evidence-based practice? *Journal of Advanced Nursing*. 47, (2004), 81–90.

[18] de Souza, L.B. 2009. Trends and approaches in lean healthcare", Leadership in Health Services. *Leadership in Health Services*. 22, 2 (2009), 121–139.

[19] UX Passion: 2017. *https://www.uxpassion.com*.

[20] 2017. *A government perspective: Tech Trends 2017*.

[21] 2010. *Patient protection and affordable care act*. Public Law 111(148), 1.