Metodologias Ágeis na Administração Pública: Uma Revisão Sistemática da Literatura

Isaque Vacari[1], Rafael Prikladnicki[2]

[1]Laboratório de Software Livre – Embrapa Informática Agropecuária
Campinas/SP, Brasil.

[2]Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre/RS, Brasil.

isaque.vacari@embrapa.br, rafaelp@pucrs.br

Abstract. *In the context of the government, public administration (PA) have undergone a significant process of modernization to meet current and emerging society commitments, but also improve the quality of public service. This fact directly implies the increased demand for software products and indirectly implies in the search for better approaches and processes to develop them or acquire them. In a modern economy, the rapidly changing market conditions and new challenges arise without warning. If for many years the software development was guided by prescriptive processes, which were designed to bring order to the chaos, providing a reasonably effective for project teams script, nowadays we observe the increasing introduction of adaptive software development processes, focused on the product and the people who develop it and recommended for environments where requirements are volatile. Agile methodologies fit in this context and has gradually attracted the attention of the Brazilian PA. Therein, the aim of this paper is to present the results of a systematic review on the use of agile methodologies in PA, presenting evidence about its benefits, threats and motivation, as well as recommendations for its use.*

Resumo. *No âmbito do governo, as administrações públicas (APs) têm passado por um significativo processo de modernização para atender os compromissos atuais e emergentes da sociedade, como também melhorar a qualidade dos serviços públicos prestados. Esta realidade implica diretamente no aumento da demanda por produtos de software e indiretamente implica na busca por melhores abordagens e processos para desenvolvê-los ou adquiri-los. Em uma economia moderna, as condições de mercado mudam rapidamente e novos desafios surgem sem aviso. Se durante muitos anos o desenvolvimento de software foi guiado por processos prescritivos, que tinham por finalidade colocar ordem no caos, proporcionando um roteiro razoavelmente eficaz para as equipes de projetos, nos dias atuais observa-se a introdução cada vez maior de processos adaptativos de desenvolvimento de software, focados no produto e nas pessoas que o desenvolvem e recomendados para ambientes onde os requisitos são voláteis. As metodologias ágeis se encaixam neste contexto e tem gradativamente despertado o interesse da AP brasileira. Neste sentido, o objetivo deste artigo é apresentar os resultados de uma revisão sistemática sobre o uso de metodologias ágeis na AP, apresentando evidências sobre seus benefícios, motivação e ameaças, além de recomendações para seu uso.*

1. Introdução

Desde o estabelecimento da Engenharia de Software (ES), o governo sempre buscou implantar padrões para estruturar o desenvolvimento de software no setor público, inspirado inicialmente em modelos prescritivos (principalmente no Modelo Cascata), que possuem em sua estrutura uma ordem formal de elementos do processo e um fluxo de trabalho que descreve como cada um destes elementos se relaciona uns com os outros, sendo sua principal característica a busca pela estrutura e ordem [Pressman 2011]. Por exemplo, o *United States Department of Defense* (DoD) institui os padrões MIL-STD-167, DOD-STD-2167 e DOD-STD-2167A entre 1974 e 1998 [McDonald 2010]. Da mesma maneira, o governo do Reino Unido determinou a adoção do padrão *Structured Systems Analysis and Design Method* (SSADM) de 1981 até meados da década de 2000 [Middleton 1999], enquanto V-Modell e suas variantes têm sido obrigatório para projetos de software no governo da Alemanha desde o início da década de 1980 [O'Connor *et al.* 2011] [Kuhrmann *et al.* 2006]. O resultado da abordagem SSADM foi percebida como prescritiva, onerosa e de difícil aplicação, assim como, mostrou dificuldade em lidar com a incerteza inerente de projetos de ES, comunicação com o usuário, desenvolvimento pessoal e não refletiu a forma como as pessoas trabalhavam na prática [Middleton 1999]. Igualmente, o DoD enfrentou diversas falhas de projeto que ocasionaram o desuso dos padrões MIL-STD-167 e DOD-STD-2167 [McDonald 2010].

Assim constatou-se que abordagens baseadas em modelos prescritivos não eram a melhor maneira de desenvolver software para a maioria dos projetos do setor público [McDonald 2010] [Middleton 1999]. Para resolver este problema, novos modelos de desenvolvimento de software, com base no modelo adaptativo, incluindo DOD-STD-2167A [McDonald 2010] e V-Modell XT [O'Connor *et al.* 2011] [Kuhrmann *et al.* 2006] foram propostos pelo governo. Entretanto, com o crescimento exponencial do mercado, a indústria de software não estava mais disposta a implementar procedimentos caros e talvez tecnicamente falhos para satisfazer as necessidades de um cliente importante, mas não mais único [McDonald 2010]. Uma das respostas da indústria de software para lidar com as limitações de projetos baseados em abordagens prescritivas de ES, típica de projetos de governo, foi dada em fevereiro de 2001, quando um grupo de dezessete renomados desenvolvedores, autores e consultores da área de software, praticantes de *Dynamic Systems Development Method*, *Extreme Programming* (XP), Scrum e *Feature Driven Development* se reuniram para discutir sobre suas experiências de trabalho e pontos em comum. O resultado do encontro deu origem ao manifesto ágil [Beck *et al.* 2001], uma declaração com quatro valores e doze princípios que são, na visão de seus proponentes, determinantes para entender o desenvolvimento de software como um processo criativo, adaptativo, e movido a incertezas.

Desde então, cada vez mais, o governo tem renunciado as suas formas de trabalho e tem adotado métodos e práticas de desenvolvimento de software alinhadas aos princípios do manifesto ágil. Por exemplo, em julho de 2012, órgãos fiscalizadores dos governos dos Estados Unidos (EUA) e do Reino Unido (UK) introduziram relatórios e diretrizes para o uso de práticas ágeis em projetos de desenvolvimento financiados pelo governo. Tanto o *Government Accountability Office* dos EUA quanto o *National Audit Office* do Reino Unido, recomendaram o uso de metodologias ágeis (MA) para desenvolvimento de software nos departamentos governamentais [GAO 2012] [NAO 2012]. No Brasil, em agosto de 2013, o Tribunal de Contas da União (TCU) emitiu na forma de acórdão o resultado de uma auditoria acerca do uso de MA em contratações para desenvolvimento

de software pela AP Federal, que tem gradativamente despertado o interesse da AP em todo o Brasil [TCU 2013].

Neste sentido, o objetivo deste artigo é apresentar um estudo pioneiro sobre a utilização de MA na Administração Pública (AP), apresentando evidências científicas sobre seus benefícios, motivação e ameaças, além de recomendações para seu uso, formando assim a base teórica sobre o tema. Para isso, planejou-se e executou-se um estudo secundário do tipo Revisão Sistemática da Literatura (RSL). Até o momento, nenhuma RSL sobre MA na AP foi publicada. Isto significa que profissionais da indústria de software, servidores públicos e pesquisadores precisam recorrer de maneira exploratória a livros e artigos, a fim de obter uma visão geral sobre o tema. Espera-se que este estudo seja útil para todos os grupos, deixando mais claro que desenvolvimento ágil de software na AP tem sido suportado por estudos científicos.

Este artigo está organizado da seguinte maneira. A Seção 2 mostra o detalhamento da RSL. A Seção 3 apresenta a execução da revisão sistemática, enquanto a Seção 4 mostra uma análise dos resultados. Finalmente, a Seção 6 apresenta as considerações finais.

2. Método de Pesquisa

O principal objetivo deste estudo foi reunir e analisar evidências científicas sobre o desenvolvimento ágil de software na AP através da execução de uma RSL, conforme as recomendações fornecidas por Kitchenham & Charters [Kitchenham and Charters 2007], incluindo a elaboração do protocolo de pesquisa e a questão de pesquisa que a revisão se propõe a abordar, assim como, os critérios de seleção de estudos.

2.1. Questão de pesquisa

Esta RSL foi iniciada com a seguinte questão de pesquisa.

O que se sabe sobre o uso de metodologias ágeis em projetos de desenvolvimento de software na Administração Pública?

2.2. Estrutura da pergunta

Para subsidiar a atividade de construção da estratégia de busca, o escopo geral do estudo foi definido como sendo:

- **População:** Administração Pública (AP).
- **Intervenção:** Artigos científicos que abordam o desenvolvimento de software na AP.
- **Saídas:** Os modelos, processos e métodos de Engenharia de Software adotados na AP, assim como, os resultados de sua utilização, incluindo os problemas, desafios, lições aprendidas, benefícios e recomendações.

2.2. Estratégia de busca

Os termos de busca foram selecionados a partir de um estudo preliminar formado por um conjunto candidato de artigos, sendo organizados em duas categorias principais: aqueles relacionados com a dimensão da AP e aqueles relacionados com a dimensão de Desenvolvimento de Software. A Tabela 1 apresenta as palavras-chave utilizadas em cada categoria para recuperar automaticamente estudos escritos em inglês.

Tabela 1. Palavras-chave utilizadas por categoria.

Referência	Categoria	Palavras-chave
A	**Administração Pública**	Government (1) Public sector (2) Public administration (3) Public organization (4)
B	**Desenvolvimento de Software**	Software development life cycle (5) Software development methodology (6) Software development process (7) Software development projects (8) Software process (9) Unified process (10) Rational unified process (11) RUP (12) Microsoft solutions framework (13) Agile methodologies (14) Agile methods (15) Agile principles (16) Agile process (17) Agile software development (18) Extreme programming (19) Lean software development (20)

A estratégia de busca combinou os termos das categorias A e B com o operador booleano "AND", sendo que os termos contidos em cada categoria foram combinados com o operador "OR". A Categoria B possui mais termos e reflete o fato de haver muitas variações sobre a indexação dos artigos. Além disso, não se tinha conhecimento da quantidade de evidências que seriam encontradas sobre desenvolvimento de software na AP. Genericamente, a frase de busca foi definida como sendo:

(1 OR 2 OR 3 OR 4) AND (5 OR 6 OR 7 OR 8 OR 9 OR 10 OR 11 OR 12 OR 13 OR 14 OR 15 OR 16 OR 17 OR 18 OR 19 OR 20)

2.3. Bases de dados

A abordagem inicial definia que as seguintes bases de dados científicas deveriam ser consultadas: ACM Digital Library, Bielefeld Academic Search Engine, ScienceDirect, Engineering Village, IEEEXplore, Scopus, SpringerLink, Web of Knowledge e Wiley Online Library. Além dessas, outras fontes de evidências científicas foram incluídas no estudo para pesquisa manual de estudos em português: Bases de Dados da Pesquisa Agropecuária, Biblioteca Digital Brasileira de Computação e Workshop Brasileiro de Métodos Ágeis.

2.4. Critérios de seleção

Os critérios de seleção de estudos destinam-se a identificar os estudos primários que fornecem evidência científica direta sobre a questão de pesquisa. Para incluir um estudo na análise, os seguintes critérios foram adotados: i) o estudo deveria relatar a experiência do uso de MA no desenvolvimento de software na AP; ii) o estudo deveria ter sido escrito em inglês ou português e iii) o estudo deveria estar disponível em texto completo para leitura e extração dos dados.

3. Execução da Revisão Sistemática

Para gerenciar o grande número de referências que foram obtidos através da pesquisa bibliográfica, a ferramenta Start 2.0 (Zamboni, 2010) foi adotada para apoiar a execução desta RSL. Os estudos selecionados para análise em profundidade foram obtidos a partir de quatro etapas, conforme Figura 1.

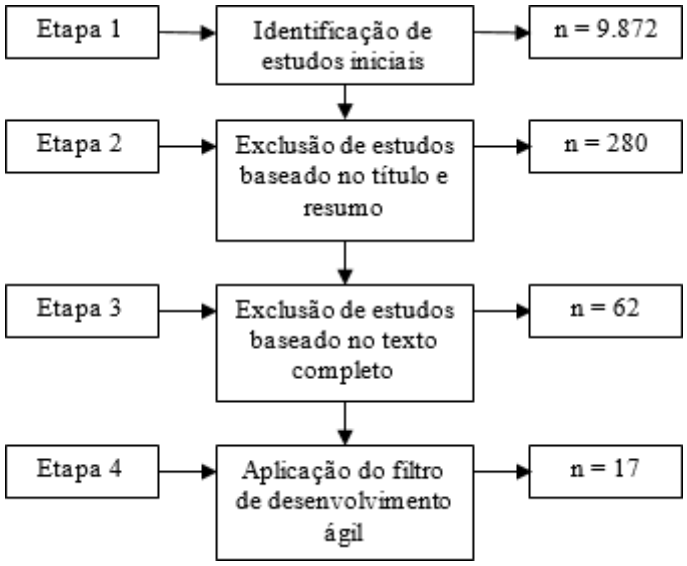

Figura 1. Etapas do processo de seleção de estudos.

Na primeira etapa, executou-se a frase de busca em cada base de dados selecionada, sendo o resultado da pesquisa bibliográfica catalogado na ferramenta Start. A segunda etapa foi baseada na leitura do título e do resumo dos trabalhos. Nesta etapa os artigos foram classificados em três categorias:

- **[Inc]** indica que o estudo está relacionado com desenvolvimento de software (DS) na AP.
- **[Exc]** indica que o estudo não está relacionado com DS na AP.
- **[Dup]** indica que o estudo está duplicado, ou seja, repetido com outros estudos.

Todos os estudos da segunda etapa contidos nas categorias [Exc] e [Dup] foram excluídos. Na terceira etapa os trabalhos da categoria [Inc] foram analisados com mais cautela através da leitura do texto completo (introdução, conclusão, e partes específicas associadas com a contribuição principal). Nesta etapa, foi incluída uma nova categoria de exclusão de trabalhos para indicar que o texto completo do estudo não está disponível para leitura [Ntc]. Todos os estudos da terceira etapa contidos nas categorias [Exc], [Dup] e [Ntc] foram excluídos. A inclusão da terceira etapa foi adequada, pois em alguns casos, a leitura do título e resumo não foi suficiente para classificar cada artigo corretamente. Desta forma, um subconjunto de documentos relacionados com desenvolvimento de software na AP contidos na categoria [Inc] foi selecionado para a etapa seguinte.

Na quarta etapa, foi incluído um classificador para indicar que o estudo está relacionado com desenvolvimento ágil de software na AP. Somente os estudos classificados como "ágeis" foram selecionados para a fase seguinte de extração de dados e análise em profundidade. Os trabalhos analisados foram catalogados na ferramenta Start 2.0 de acordo com três categorias gerais de informação, incluindo informações gerais, informações relacionadas com a organização da pesquisa e com o conteúdo da pesquisa.

4. Análise dos Resultados

Na primeira etapa, um total de 9.872 artigos foram encontrados. Após a triagem inicial, realizada na segunda etapa, um total de 280 artigos foram selecionados para a terceira etapa. Após a segunda triagem, realizada na terceira etapa, um total de 62 artigos foram selecionados para uma terceira triagem. Finalmente, dos 62 artigos encontrados, 17 estudos sobre desenvolvimento ágil de software na AP foram selecionados para uma análise em profundidade, conforme Tabela 2. Foi observado que a falta de terminologia para desenvolvimento de software na AP e a baixa precisão dos mecanismos de buscas resultou em uma grande quantidade de estudos iniciais. Embora a estratégia de busca tenha retornado um número maior de artigos para seleção, somente alguns deles foram relevantes para responder a questão de pesquisa.

Tabela 2. Conjunto final de artigos para analise em profundidade.

ID	Título	Autoria	Ano	BD
[P01]	A case study: Introducing eXtreme programming in a US government system development project	A. Fruhling, P. McDonald, and C. Dunbar	2008	Scopus
[P02]	Staying agile in government software projects	B. Upender	2005	Scopus
[P03]	Adoção de métodos ágeis em uma Instituição Pública de grande porte - um estudo de caso	C. de O. Melo, and G.R.M. Ferreira	2010	WBMA
[P04]	The FBI gets agile	C. Fulgham; J. Johnson; M. Crandall; L. Jackson; N. Burrows	2011	Scopus
[P05]	Collaborative development of public information systems: A case study of "Sambruk" e-services development	C.-O. Olsson, and A. Öhrwall Rönnbäck	2010	Scopus
[P06]	Making agile development work in a government contracting environment-measuring velocity with earned value	G.B. Alleman, and M. Henderson	2003	IEEE
[P07]	Agile development in a bureaucratic arena - A case study experience	H. Berger	2007	Scopus
[P08]	An industrial case study for Scrum adoption	H. Hajjdiab, A.S. Taleb, and J. Ali	2012	Scopus
[P09]	Army simulation program balances agile and traditional methods with success	J. Surdu, and D.J. Parsons	2006	Scopus
[P10]	A Case Study on the Adoption of Measurable Agile Software Development Process	M. Iliev, I. Krasteva, and S. Ilieva	2009	BASE
[P11]	Extreme Programming by example	M. Pedroso Jr, M.C. Visoli, and J.F.G. Antunes	2002	BDPA
[P12]	Lessons learned using agile methods on large defense contracts	P.E. McMahon	2006	Scopus
[P13]	Evolving to a "lighter" software process: a case study	R.J. Moore	2001	IEEE
[P14]	Is Agile the Answer? The Case of UK Universal Credit	R. Michaelson	2013	Springer
[P15]	Agile software development under university-government cooperation	S. Kaneda	2006	Scopus
[P16]	Exploring XP for scientific research	W.A. Wood, and W.L. Kleb	2003	Scopus
[P17]	Agile metrics at the Israeli Air Force	Y. Dubinsky, D. Talby, O. Hazzan, and A. Keren	2005	Scopus

O processo de classificação e análise de artigos com base em alguns critérios pode ser subjetivo. Para minimizar esta limitação, uma abordagem em quatro etapas foi planejada para seleção de artigos, explicada na seção 3, e uma outra abordagem de revisão de todos os artigos selecionados foi planejada para análise dos estudos. Todos os estudos foram lidos pelo menos três vezes pelo mesmo pesquisador, em momentos diferentes, visando buscar a estabilidade da análise realizada.

Uma vez que o principal interesse é o estudo do uso de MA na Administração Pública (AP), a análise dos estudos incidiu sobre essa temática, sendo organizada em algumas dimensões, mostradas na sequência.

4.1. Alguns motivos e benefícios das metodologias ágeis

Uma das maiores motivações para a adoção de MA são os benefícios que elas podem trazer para o setor público, como uma resposta ao histórico de fracasso de projetos de TI no governo [P02] [P04] [P08] [P11] [P14], incluindo: (1) uma entrega rápida de valor ao cliente; (2) uma maior colaboração entre TI e negócios; (3) uma maior satisfação do cliente. Além disso, parece haver um fato novo: (4) uma elevação da moral da equipe de TI do governo, reduzindo a dependência de empresas contratadas.

Entrega rápida de valor ao cliente

Embora a velocidade para entrega de um produto para o cliente não seja um verdadeiro atributo de qualidade no sentido técnico, é uma medida de qualidade do ponto de vista de negócio [Pressman 2011]. Do ponto de vista de negócio, dividir o software em um conjunto mínimo de funcionalidades úteis e entregar cada conjunto no seu devido tempo, o de mais alto valor primeiro, tem ajudado os clientes a fazerem melhor uma parte útil de seus trabalhos de maneira antecipada [P02] [P09] [P12] [P13]; bem como, quando esses conjuntos de funcionalidades úteis começam a produzir mais cedo um retorno de investimento, sistemas anteriores tem sido descontinuados, gerando uma economia [P05]; como também; enquanto mais valor são entregues ao cliente mais cedo, ganhos em produtividade com reduções equivalentes no custo tem acontecido [P04].

Colaboração entre TI e negócios

Desde sempre tem se falado da necessidade de um alinhamento maior de TI aos negócios [Pressman 2011]. Se os negócios estão, cada vez, mais mutáveis e incertos, então o trabalho de TI para estar mais alinhada só aumenta. Para isso, é necessário trabalhar em estreita colaboração com os parceiros de negócios, aproximando desenvolvedores e clientes [P01] [P02] [P04] [P07] [P09] [P10] [P17] [P05], entregando o maior valor de negócio possível [P02] [P09] [P12] [P13], no menor tempo [P02] [P04] [P09] [P12] [P13] e custo possíveis [P04] [P07], ajudando-os a empregar o sistema de forma efetiva e entregando mais e melhores funcionalidades ao longo do tempo [P04].

Satisfação do cliente

O resultado de um alinhamento maior de TI aos negócios, bem como, de uma entrega rápida de valor ao cliente, com base no *feedback* do cliente [P10] tem sido uma maior satisfação do cliente com o produto de software desenvolvido [P02] [P03] [P10], tal como, um aumento na confiança da equipe [P17]. Ao tornar o processo de desenvolvimento mais aberto para o cliente, a implementação de novas funcionalidades tem se tornado algo previsível, transparente e mais aderente as necessidades do cliente [P01] [P03] [P04] [P09] [P10] [P16].

Moral da equipe e redução da dependência de empresas contratadas

Ainda que o assunto de contratação do desenvolvimento de software no governo não seja novo, o papel do governo nesse processo tem sido um dos temas mais discutidos nos últimos anos [P03] [P04] [Heil 2010]. Dentre os aspectos mais relevantes do governo está a sua capacidade em lidar com novas metodologias e tecnologias para o desenvolvimento de software [Heil 2010]. Embora a participação de empresas da indústria de software seja

vista com “bons olhos”, cabe ao governo, buscar a eficiência e a liderança na execução do desenvolvimento de software [Heil 2010]. Quando o governo tornou-se totalmente dependente de empresas da indústria de software, então constatou-se uma desmotivação nos empregados de governo, por trabalhar grande parte do tempo em tarefas burocráticas de gerenciamento de empresas ou em atividades não relacionadas a implementação de software em si [P03]. Por outro lado, quando o governo assumiu a liderança e participou de todas as etapas do desenvolvimento de software com empresas contratadas, bem como, quando o governo adotou novos e mais modernos métodos de desenvolvimento de software (que inclui MA) houve uma melhoria significativa na moral da equipe [P04] [P11] [P17] e na disciplina de ES [P03], o que tem reduzido a dependência de empresas privadas [P03] [P04].

Outros benefícios das metodologias ágeis

Secundariamente, outros benefícios têm sido encontrados, incluindo melhorias no(a): (1) comunicação entre os membros da equipe de desenvolvimento, bem como, entre desenvolvedores e clientes [P01] [P02] [P04] [P09]; (2) aprendizado de novas tecnologias [P03]; (3) qualidade do produto [P16]; (4) visibilidade do projeto [P02] [P12] [P17]; (5) produtividade das equipes [P03] [P16]; (6) redução de custos [P04] [P07]; (7) capacidade de gerenciar mudanças e prioridades [P04] [P05] [P17] e (8) conformidade com exigências burocráticas de governo [P06].

4.2. Algumas dificuldades na adoção de metodologias ágeis

Em alguns casos, a imagem bastante otimista no nível teórico dos MA pode ser contraposta por uma realidade prática dominada por desafios, dificuldades e problemas concretos. Na sequência, algumas dificuldades na adoção de MA na AP são apresentadas.

O impacto da cultura organizacional

Segundo Kent Beck [Beck 1999], a maior barreira para o sucesso de XP é a cultura organizacional. Isso é uma verdade para as MA como um todo, e não apenas para XP. De acordo com o relatório “*8th State of Agile Development Survey Results*” [VersionOne 2013], capacidade de mudança da cultura organizacional continua sendo a maior barreira para adoção de MA, com mais da metade dos entrevistados citando esse aspecto como maior problema. Qualquer projeto executado em estruturas hierárquicas burocráticas, típicas de governo, terá conflitos com equipes que preferem práticas de trabalho flexíveis e colaborativas [P01] [P02] [P03] [P07] [P08] [P14] [P15] [P17]. Por exemplo, no caso do(a) *UK Regional Government Departament* [P07], aspectos inerentes da cultura organizacional impediram que as partes interessadas conseguissem cooperar e colaborar com a equipe de desenvolvimento, gerando um impacto negativo sobre o progresso do projeto e a confiança da equipe. A “cultura da culpa”, diminui a capacidade para a tomada de decisão sobre os objetivos do negócio, a qual afetou o trabalho dos desenvolvedores que necessitavam de uma priorização das atividades para cumprir os prazos de desenvolvimento; com isso, o progresso do projeto foi prejudicado com atrasados no cronograma, que contribuíram para situações de conflitos e perda da confiança no trabalho colaborativo.

Outra cultura que não contribui para as MA é o enraizamento do planejamento prévio e detalhado no setor público. Embora, a adoção de MA tenha sido bem sucedida no(a) *United States Strategic Command* [P01], várias partes interessadas continuavam acreditando que com o planejamento prévio e detalhado melhores resultados teriam sido

alcançados. Isso aconteceu apesar do fato de que as percepções da qualidade do produto resultante eram elevadas, e que a entrega frequente de novas funcionalidades, a cada duas ou três semanas em vez de dois meses, foi muito apreciada. Ou seja, qualquer projeto que tente apontar a direção certa logo de "cara" terá conflitos com equipes que preferem ir acertando a direção continuamente [Beck 1999].

Uma cultura pronta para mudanças, disposta a encarar o novo, com coragem, descobrindo novos caminhos e novas soluções é a condição ideal para as MA começar [P01] [P07]. Porém, em alguns casos, organizações públicas preferem abordagens que estão mais estreitamente alinhadas aos seus processos existentes, permanecendo como está, desertando da transformação [P08]. Assim, em muitas organizações as pessoas sabem e dominam apenas o que elas estão acostumadas. Por isso, a personalização de MA, bem como, a não experimentação de determinadas práticas ágeis onde elas não são compatíveis com a cultura organizacional, não é incomum no setor público [P01]. Por outro lado, uma abordagem mais flexível, na qual a equipe aprende MA e encontra respostas para os problemas culturais da organização a partir da experimentação do método em projetos pilotos, tem alcançado impactos positivos [P03] [P16].

A falta de conhecimento e experiência com metodologias ágeis

A falta de conhecimento e experiência dos servidores públicos (incluindo desenvolvedores, gerentes, clientes e usuários finais) e de empresas contratadas da indústria de software para executar o desenvolvimento de software de maneira diferente, de modo a entregar serviços melhores e mais ágeis tem sido uma questão latente nas iniciativas de adoção de MA na AP, conforme mostrou os estudos [P01] [P02] [P03] [P08] [P11] [P14] [P17]. Há várias características sobre MA que são novas e desconhecidas para muitos membros da equipe [P03] [P11]. Elas têm exigido uma mentalidade muito diferente do que as pessoas estão acostumadas [P17]. A experiência de abordagens baseadas em modelos prescritivos é completamente diferente do que realizar reuniões diárias, trabalhar com *timebox*, entregar em pequenos incrementos de software, bem como, manter histórias de usuário e *backlogs* de produto [P17]. Por exemplo, dificuldades em convencer os clientes a realizar publicações parciais do sistema em produção, mesmo que agregassem valor ao negócio, foram identificadas [P03].

Um risco comum causado pela falta de experiência com MA é que a equipe quando encontra uma prática ágil difícil de aplicar, ela tenta alterar a MA para seu contexto específico (o que pode desvirtuar a essência da metodologia), em vez de aprender mais sobre os benefícios da prática visando mudar a forma como a equipe trabalha. Por isso, é importante ter o acompanhamento de pessoas experientes nos estágios iniciais de adoção de MA; caso contrário, as iniciativas para sua adoção irão falhar, foi o que aconteceu em um departamento de TI do governo dos Emirados Árabes Unidos [P08].

O pouco ou nenhum comprometimento das partes interessadas

Projetos de governo precisam do apoio e participação ativa das partes interessadas [P07]. Quando isso aconteceu, as melhores soluções de software foram criadas, inclusive com uma maior aceitação na organização pública e na sociedade [P07]. Normalmente, as partes interessadas precisam investir tempo e recursos para tornarem um projeto viável; porém, muitos projetos podem tornarem-se inviáveis quando as partes interessadas não são aliviadas de seus trabalhos diários para participar ativamente do projeto [P07].

O enraizamento de abordagens prescritivas no setor público

Embora vários aspectos do desenvolvimento adaptativo têm sido defendidos e valorizados pela comunidade de ES durante anos, ainda existe um viés em direção a abordagens prescritivas no setor público, incluindo Modelo Cascata, *Big Design Up Front* (BDUF) e *big bang*, conforme mencionado nos estudos [P01] [P04] [P15] [P17] [P19]. A abordagem Cascata é apropriada para alguns projetos da área de Engenharia Civil, que são monolíticos por natureza, tais como a construção de prédios [G06], mas quando aplicados em projetos de desenvolvimento de software tenderão para o que Kent Beck [Beck 1999] nomeou de BDUF, que busca a precisão e a lógica perfeita na definição do escopo do projeto antes de iniciar o desenvolvimento do software, sendo recomendado apenas para situações onde os requisitos são razoavelmente estáveis [Pressman 2011]. Essa situação foi evidenciada no caso do projeto *Sentinel* do *Federal Bureau of Investigation* (FBI) [P04], onde o plano inicial do projeto mostrou-se irrealista, bem como, o desenvolvimento e a entrega de software executadas em grandes períodos de tempo (*big bang*), por empresas contratadas, não conseguiu atender as necessidades dos usuários. A solução encontrada pelo FBI envolveu a participação de desenvolvedores de software de governo em atividades de codificação, bem como, uma mudança na estratégia de gerenciamento do projeto; o FBI assumiu literalmente a liderança do projeto e adotou o método Scrum para ajuda-lo nos aspectos gerenciais do produto de software; ele substituiu um extenso plano de projeto por um *backlog* de produto (priorizado e organizado em histórias de usuário), assim como, incluiu o desenvolvimento incremental em tempos curtos de entrega com maior *feedback* e envolvimento dos usuários.

O perigo de contratos e contratações

O governo precisa saber exatamente quando o projeto será entregue e quanto custará aos cofres públicos. Para isso, durante anos, a única estratégia adotada pelo governo foi ter como base um escopo suficientemente detalhado e documentado com estimativas mais próximas da certeza para tal escopo [P01] [P03] [P04]. Com essa abordagem, o governo procurou prever exatamente o que receberia, quando e com que preço. A princípio, essa abordagem parecia simplificar o processo de compras para contratação de empresas da indústria de software [P01] [P04]. Porém, esse cenário previsível e com poucas mudanças, na maioria dos casos, não tem se configurado no setor público; pelo contrário, a natureza, cada vez mais, dinâmica e complexa dos negócios e da sociedade provocam mudanças no setor público, sendo desejável que projetos absorvam essas mudanças naturalmente [P04].

Por conta disso, a previsibilidade sobre o escopo é inviável na maioria dos casos, e fixar o escopo pode prejudicar o governo em situações de mudanças em seus processos de negócio. A insistência nessa abordagem tem aumentado o risco de enfrentar e resolver o problema errado, além de inserir processos onerosos na tentativa de fechar o escopo, os quais dificultam a absorção de mudanças [P04]. Por esses motivos, diversos projetos de "escopo fixo" no setor público fracassaram, onde soluções foram entregues sem resolver os problemas reais dos usuários, com pouca ou nenhuma contribuição para os objetivos propostos do projeto [McDonald 2010] [Middleton 1999] [P04].

Uma solução interessante para o problema de contratos de "escopo fixo", conhecida como contrato de "escopo negociável", quando combinada com MA, tem emergido no setor público com resultados positivos [P04]. Nesse modelo, o escopo é negociado e discutido diversas vezes ao longo do projeto; ele não está vinculado ao contrato, assim, não há risco da empresa contratada deixar de cumprir com o contrato por um erro de interpretação da

equipe ou alterações no escopo efetuadas pelo governo (cliente) ao longo do projeto. No entanto, projetos ágeis com escopo negociável dependem de decisões com base na confiança mútua, sendo bem adequados para projetos internos [Beck 1999]. Assim, contratar o desenvolvimento ágil quando a solução está sob a liderança de empresas contratadas externamente parece não proporcionar benefícios reais [Beck 1999] [P03] [P04]; pelo contrário, aumenta a dependência do governo em relação a indústria de software, bem como, diminui a competência do governo em manter sistemas críticos internamente, o qual deveria deter a competência em lidar com as metodologias e tecnologias utilizadas, tal como, a compreensão associada ao impacto em custo e cronograma [Beck 1999] [Heil 2010].

5. Limitações da Revisão Sistemática da Literatura

A RSL descrita neste trabalho contribuiu para identificar estudos dentro do domínio pesquisado. Contudo, como qualquer outro método, foram encontradas algumas limitações. Primeiramente, a escolha das bases de dados para consulta procurou abranger o máximo das principais alternativas científicas disponíveis. Entretanto, é possível que outras fontes de publicações não utilizadas neste estudo também contenham artigos sobre desenvolvimento de software na AP. Portanto, não é possível garantir a cobertura total de artigos científicos sobre esse assunto. Em segundo lugar, relatórios de governo não foram incluídos. Finalmente, a leitura dos artigos foi realizada por um único pesquisador.

6. Considerações Finais

Uma conclusão desse estudo é que MA têm produzido resultados melhores do que é possível alcançar adotando-se modelos prescritivos em ambientes de governo [P01] [P03] [P04] [P09] [P11] [P16] [P17]. O principal argumento é que modelos prescritivos não fornecem a visibilidade do produto de software ou resultados concretos até o final do ciclo de vida do projeto, sendo considerado adequado para situações onde os requisitos são estáveis [P07]. Consequentemente, MA evoluíram para proporcionar uma abordagem de desenvolvimento mais flexível e adequada para situações onde os requisitos são voláteis e incertos [P07]. O sucesso de MA deve-se principalmente ao fato de que os resultados priorizados são entregues mais cedo, e em incrementos muito menores do que em projetos prescritivos, onde cada etapa do projeto deve ser totalmente concluída antes de prosseguir para a etapa seguinte [Wernham, 2012]. Secundariamente, há um elemento de aprendizagem inserido em cada ciclo de entrega e uma elevação na moral da equipe [P03] [P04].

Outrossim, uma forte recomendação dessa pesquisa é que, para ser adaptativo, os governos precisam criar as suas próprias capacidades organizacionais e técnicas para atuar em estreita colaboração, em todas as etapas do ciclo de vida do projeto, com as empresas contratadas da indústria de software, e não apenas monitorarem e verificarem os resultados finais das tarefas realizadas. Os governos deveriam parar de contratar o desenvolvimento de software por inteiro com grandes etapas de entrega, bem como, deveriam parar de transferir a responsabilidade do desempenho e resultados de projetos para as empresas contratadas [P04] [P09]. Além disso, a opção do governo em transferir completamente o trabalho de desenvolvimento para a indústria raramente tem sido uma alternativa viável [Heil 2010] [P04] [P09]. O impacto de custo e cronograma tem feito com que as únicas opções de mitigação de risco sejam: aumentar significativamente os recursos financeiros, atrasar significativamente o cronograma, reduzir ou eliminar significativamente as funcionalidades previstas, ou então, cancelar o projeto [Heil 2010].

Para reverter esse quadro, os governos precisam assumir a liderança e necessitam aumentar a sua participação em projetos de desenvolvimento de software, principalmente os de missão crítica para a AP. Por outro lado, mais pesquisas para compreender a dinâmica jurídica para lidar com as deficiências dos modelos de contratos tradicionais e encontrar abordagens adequadas para unir o governo (contratante), o governo (desenvolvedor) e empresas contratadas são necessárias.

Trabalhos futuros

Finalmente, uma vez que MA não têm sido testadas e exploradas suficientemente no setor público, assim como, dado o conjunto restrito de evidências encontradas para extrair resultados conclusivos, mas ao mesmo tempo a existência de resultados promissores do uso de MA na AP, existe uma oportunidade para pesquisadores atuarem em ambientes de governo para realizarem estudos empíricos para entender como elas são executadas na prática e que efeitos geram. Esta e outras oportunidades serão exploradas na continuidade desta pesquisa, que tem como objetivo entender e ampliar o que se sabe sobre a adoção de MA no setor público em benefício do desenvolvimento de software na AP brasileira.

6. Referências

Beck, K. (1999) “Extreme programming explained: embrace change”. Boston: Addison-Wesley, 190 p. il. (The XP series)

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. (2001) “Manifesto for agile software development”. http://agilemanifesto.org.

GAO (2012) “GAO: Software Development - Effective Practices and Federal Challenges in Applying Agile Methods”, United States Government Accountability Office, 34p.

Heil, J. W. (2010) "Addressing the Challenges of Software Growth and Rapidly Evolving Software Technologies," Naval Engineers Journal, vol. 122, pp. 45-58.

Kitchenham, B.A. and Charters, S. (2007) “Guidelines for Performing Systematic Literature Reviews in Software Engineering” Technical Report EBSE-2007-01.

Kuhrmann, M., Niebuhr, D. and Rausch, A. (2006) "Application of the V-Modell XT - Report from a pilot project," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) vol.3840 LNCS, pp.463-473.

O‘Connor, R., Pries-Heje, J., Messnarz, R., Kuhrmann, M., Lange, C. and Schnackenburg, A. (2011) "A Survey on the Application of the V-Modell XT in German Government Agencies," in Systems, Software and Service Process Improvement. vol. 172, ed: Springer Berlin Heidelberg, pp. 49-60.

Pressman, R. S. (2011) “Engenharia de Software: Uma abordagem profissional”. Tradução de Ariovaldo Griesi. 7.ed. Porto Alegre: AMGH, 2011. 780 p. il.

McDonald, C. (2010) "From art form to engineering discipline? A history of us military software development standards, 1974-1998," IEEE Annals of the History of Computing, vol. 32, pp. 32-45.

Middleton, P. (1999) "Managing information system development in bureaucracies," Information and Software Technology, vol. 41, pp. 473-482.

NAO (2012) “NAO: Governance for Agile delivery”, National Audit Office, 35p.

TCU (2013) “TCU: Levantamento de Auditoria. Conhecimento acerca da utilização de métodos ágeis nas contratações para desenvolvimento de software pela Administração Pública Federal”, Tribunal de Contas da União, 42p.

VersionOne (2013) “8th Annual State of Agile Development Survey”. VersionOne, 2013. 14 p. disponível em: <http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>. Acesso em: 26 ago. 2014.

Wernham, B. “Agile Project Management for Government”, 1. ed. London - New York - Sydney: Maitland and Strong Publishing, 2012.

Zamboni, A. B. et al. “StArt Uma Ferramenta Computacional de Apoio à Revisão Sistemática”. In: Proc.: Congresso Brasileiro de Software (CBSoft’10), Salvador, Brazil. 2010.