

METODOLOGIAS ÁGEIS NO DESENVOLVIMENTO DE SOFTWARES: uma revisão bibliográfica

Adival José Reinert Junior
Neide Pereira de Oliveira

RESUMO

O ambiente de desenvolvimento de software está à mercê de evoluções tecnológicas e inovações constantes, gerando a exigência de produtos complexos e ou com mudanças de escopo, as metodologias tradicionais não conseguem acompanhar essas alterações, levando o desenvolvedor justificar-se pelo produto entregue muitas vezes aquém ao que foi solicitado, devido a dinâmica entre o tempo de desenvolvimento e a entrega do produto. O manifesto Ágil surgiu como uma resposta aos problemas enfrentados pela engenharia de software, sobretudo as questões de alterações de requisitos e a complexidade dos novos sistemas, assim os métodos ágeis cada vez mais ganham espaço no mercado. Nesse contexto, surge o questionamento: como as metodologias Ágeis podem auxiliar no desenvolvimento de softwares? Desse modo, o objetivo geral deste artigo é apresentar as diferenças existentes entre os métodos de desenvolvimento de software tradicionais e os métodos ágeis. E, como objetivos específicos descrever os principais problemas enfrentados, assim como as vantagens obtidas na utilização das metodologias Ágeis no desenvolvimento de softwares. As metodologias Ágeis balizam o desenvolvimento de softwares de qualidade, cumprindo prazos e aproveitando os recursos disponíveis, através de métodos capazes em cada iteração entregar um produto de qualidade que atenda as expectativas do cliente.

Palavras-chave: Engenharia de Software. Metodologias Ágeis. Metodologias Tradicionais. Scrum. XP.

1 INTRODUÇÃO

O ambiente de desenvolvimento de software sofre constantes alterações de escopos, evoluções tecnológicas e inovações, assim sendo, muitas vezes o desenvolvedor passa pelo crivo do cliente usuário que traz à tona a crítica entre o que foi solicitado e o que foi desenvolvido, tendo em vista a dinâmica entre o tempo de desenvolvimento e a entrega do produto, sendo um desafio a ser enfrentado diante, em geral, a recursos escassos que envolvem prazos e custos, que impactam diretamente na qualidade do produto.

Em 2001 foi publicado o chamado Manifesto Ágil (MANIFESTO, 2001), uma declaração pública que esclarece os princípios, valores e motivações do desenvolvimento ágil de software, com a intenção de contrapor as abordagens tradicionais de desenvolvimento de software, formulando a metodologia ágil com a proposta de uma comunicação mais eficaz entre os stakeholders (envolvidos) no projeto, com um fluxo de desenvolvimento organizado com o mínimo de desperdício de recursos.

Nesse contexto, surge o questionamento: como as metodologias Ágeis podem auxiliar no desenvolvimento de softwares?

Desse modo, o objetivo geral deste artigo é apresentar as diferenças existentes entre os métodos de desenvolvimento tradicionais e os métodos ágeis. E, como objetivos específicos descrever os principais problemas enfrentados, assim como as vantagens obtidas na utilização das metodologias Ágeis no desenvolvimento de softwares.

Este estudo está dividido, respectivamente, em desenvolvimento, composto pelo referencial teórico, metodologia da pesquisa aplicada, e os resultados e discussão dos materiais encontrados na revisão de literatura. E, finalmente em algumas considerações finais acerca do tema discutido.

2 METODOLOGIAS ÁGEIS NO DESENVOLVIMENTO DE SOFTWARES

Em um mercado em constante mudança e avanços tecnológicos o desenvolvimento de software torna-se bastante complexo para enfrentar inúmeros desafios, para Curado, Machado e Silva (2016) as metodologias tradicionais (prescritivas) de desenvolvimento apresentam problemas, enquanto as metodologias Ágeis por terem foco na entrega do produto, muitas vezes com entregas parciais e validações junto ao cliente, para só então dar continuidade ao projeto, são mais flexíveis e propensas a solucionar problemas.

A engenharia de software desempenha papel crucial no desenvolvimento de projetos e sistemas, pois abrange os estágios iniciais de especificação do sistema até a sua manutenção, considera todos os aspectos na produção do software, lembrando que software é o conjunto de programas, arquivos de configuração e a documentação do sistema. (SOMMERVILLE, 2011).

Segundo Silva, Souza e Camargo (2013), uma das tarefas fundamentais no processo de desenvolvimento de software é o gerenciamento de projetos, esses projetos possuem um ciclo de vida de desenvolvimento do sistema, sendo este dividido em quatro fases a saber: planejamento, análise, projeto e implementação.

Um dos principais fatores do sucesso de um projeto é a boa definição dos seus requisitos, os requisitos descrevem as necessidades do negócio (requisitos do negócio), o que os usuários devem fazer (requisitos dos usuários), o que o software deve fazer (requisitos funcionais) e as características que o sistema deve ter (requisitos não funcionais). (DENNIS; WIXOM; ROTH, 2014).

Trigo e Barreto (2019), afirmam que o desenvolvimento de software com o objetivo de atender aos requisitos do cliente utilizam o modelo cascata, também conhecido como water fall ou top-down, que é constituído por etapas desenvolvidas de modo sequencial, assim sendo considerado um modelo linear, pois cada fase só pode ser iniciada após a finalização da fase anterior, ou seja, uma saída é entrada para outra.

Corroboram Pressman e Maxim (2016) elencando as fases deste modelo, também conhecido como ciclo de vida clássico, que são: a comunicação que é o início do projeto com o levantamento de requisitos; planejamento com estimativas, cronogramas e acompanhamento; modelagem com a análise e projeto; construção com código e teste; e finalmente entrega que envolve a própria entrega, suporte e feedback, possuindo suporte contínuo do software construído.

A dependência existente entre as fases do modelo cascata caracteriza a sua principal desvantagem, assim sendo, somente é indicado para pequenos projetos sem à necessidade de padronizações e documentações. Outro fator a ser considerado é que uma má definição dos requisitos impacta diretamente no resultado final do software desenvolvido.

Para Sommerville (2011) o modelo incremental é uma evolução do modelo em cascata, seguindo as mesmas fases do modelo anterior, contudo se dá através de incrementos (módulos), que são pedaços do software desenvolvidos de forma linear, com as atividades de especificação, projeto, implementação e validação intercaladas, em cada incremento ou nova versão, com rápido feedback entre todas as atividades.

Cada incremento é como fosse um subsistema, e em cada subsistema é aplicado o modelo em cascata, assim a etapa inicial de análise deve estar concluída para o início do projeto, além disso o custo para integração de vários subsistemas pode ser alto, caso sejam desenvolvidos por equipes diferentes em paralelo, quanto mais complexo o sistema esse problema se agrava. (SOMMERVILLE, 2011).

Utiliza-se de um protótipo do sistema real como forma de determinar o requisito, tal protótipo deve ser de baixo custo e de rápido desenvolvimento, o uso de protótipos auxilia na exibição da interface, no processo de interação com funcionalidades e botões, facilitando a compreensão do usuário, possuindo as etapas: comunicação; modelagem – aspectos visuais do usuário; construção do protótipo; e implantação – entrega e feedback. (PRESSMAN, 2010).

Conforme Pressman (2010), dentre as desvantagens da prototipagem se destacam a obrigatoriedade do cliente estar ciente que o protótipo não é o produto final e que o mesmo será

refeito, outra questão é que os desenvolvedores não devem aproveitar na versão final o código escrito sem que haja um planejamento para tanto.

O modelo em espiral de Boehm, uma evolução dos modelos incremental e prototipação, combina prevenção e tolerância a mudanças, assim preza pela diminuição de riscos no desenvolvimento do software, cada volta no espiral percorre todas as fases do processo de software, sendo elas: comunicação, planejamento – estimativas de custos, cronograma e análise de riscos, modelagem – análise e projeto, construção – codificação e testes, entrega – entrega e feedback (SOMMERVILLE, 2011; TSUI, KARAM, 2013).

No entanto Pressman (2010) afirma que tal modelo pode ser utilizado em qualquer estágio da evolução do produto, sendo apropriado para software de grande porte, uma desvantagem a ser considerada é que se o orçamento for fixo este modelo pode se tornar um problema.

De acordo com Prikladnicki, Willi e Milani (2014) a engenharia de software está em fase de aprendizado, assim evolui de forma natural em busca de um caminho de sucesso, assim surge novos processos mais leves em contrapartida dos processos burocráticos tradicionais. Para Silva, Souza e Camargo (2013) existem várias metodologias de desenvolvimento classificadas como ágil, entre elas se destaca a metodologia SCRUM, por ser um processo de desenvolvimento software incremental em ambientes complexos, atendendo aos requisitos não claros ou que mudam com muita frequência.

O sucesso do método SCRUM segundo Sutherland e Schwaber (2017) depende das pessoas, pois como o método atua em projetos e desenvolvimento de sistemas que muitas vezes são imprevisíveis, é necessário o comprometimento das pessoas envolvidas.

Reforçando essa ideia Silva, Medeiros e Medeiros (2018) salientam que existem papéis bem definidos, cada papel possui sua necessidade específica de iteração e processo que deve ser transparente o suficiente para que cada pessoa envolvida saiba exatamente o que foi, está ou será feito.

Completando essa questão entre papéis e componentes do sistema temos: time scrum, dono do produto (product owner), mestre scrum (scrum master), time de desenvolvimento, backlog do produto, sprints, revisão da sprint (sprint review), retrospectiva da sprint, burndown chart. (SILVA; MEDEIROS; MEDEIROS, 2018).

Salientam Trigo e Barreto (2019) que aplicando o Scrum em uma microempresa o dono do produto pode ser o próprio proprietário ou algum funcionário que consiga transmitir o que a microempresa necessita.

Assim como, os sprints constituem uma entrega de software, de maior qualidade, em pequenos intervalos de tempo fixo, que em geral duram menos de um mês, o maior objetivo do sprint é entregar o máximo valor de negócio possível no menor tempo. (BESSA; ARTHAULD, 2018).

O método Scrum tem como fundamento o empirismo, onde o conhecimento vem das experiências e tomadas de decisões, que possui três pilares que são a transparência - os principais aspectos do processo estejam visíveis e padronizados para que os interessados tenham o mesmo entendimento; inspeção - detectar indesejáveis variações e precisa ser realizada, de preferência, por um especialista; e adaptação - o processo ou produto tem de ser adaptado quando o especialista identificar um desvio ou produto inaceitável. (SUTHERLAND; SCHWABER, 2017; TERLIZZI; BIANCOLINO, 2014).

O Extreme Programming ou XP (Programação Extrema) é um método ágil que atende especialmente projetos construídos por times de 2 a 10 programadores, além de ser considerado leve, de baixo risco, eficiente e flexível, possuindo quatro atividades básicas ouvir, desenhar, codificar e testar. (ALLIANCE, 2018).

Outro método ágil é o Kanban que diferente de outros métodos ágeis não possui interações, desacoplando o planejamento, priorização, desenvolvimento e entrega, deixando que cada atividade dessas possa ter sua própria cadência, possibilitando ajustar-se à realidade e necessidades demandadas pelo processo. (CURADO; MACHADO; SILVA, 2016).

Bessa e Arthauld (2018) simplificam o modo de se pensar Kanban, esta palavra japonesa significa cartão visual, o que torna-se uma vantagem, à medida que permite um controle visual das atividades existentes, em andamento ou na fila para iniciar, facilita a identificação de possíveis problemas no processo de desenvolvimento, possibilitando a atuação de uma equipe multifuncional, o estímulo de melhoria contínua (kaizen), redução dos desperdícios (lean), menos burocracia, e com atividades realizadas no tempo correto (just in time).

O presente estudo foi desenvolvido a partir de uma revisão bibliográfica sistemática, ratificando as etapas descritas por Gil (2010), que são: escolha do tema, levantamento bibliográfico preliminar, definição do problema, planejamento, busca das fontes bibliográficas, leitura do material (exploratória, seletiva, analítica, e interpretativa), fichamento, organização lógica do assunto, e redação do texto.

A pesquisa foi realizada utilizando os seguintes critérios de inclusão: busca on-line de artigos científicos recentes, capítulos de livros, revistas e periódicos, com datas de publicações

entre os anos de 2010 a 2020, sobre a temática ou correlacionado a mesma, publicados em plataforma virtual, e publicações não digitais seguindo os mesmos critérios.

Após a análise crítica dos trabalhos selecionados percebe-se que o desenvolvimento de software requer a escolha da melhor metodologia, sendo àquela que atenda às necessidades e requisitos dados pelos usuários.

Consonantes Bessa e Arthauld (2018) destacam que as metodologias ágeis são adaptativas ao invés de predeterminantes como as tradicionais, são orientadas a pessoas e não a processos, enquanto as metodologias tradicionais ainda no início de desenvolvimento tentam prever detalhes do processo, não sendo apropriadas a problemas de mudanças no projeto, por exemplo, os atrasos na entrega do projeto.

Michels e Ferreira (2013), através de uma revisão bibliográfica sistemática com cerca de 209 estudos, destacam que atualmente a gestão ágil se manifesta sobremaneira no desenvolvimento de produtos inovadores, e detrimento do uso dos modelos prescritivos.

Ainda esses autores inter-relacionam os princípios ágeis e desenvolvimento desses novos produtos, seguindo esses valores como premissas de sucesso, onde os indivíduos e interações são mais importantes que processos e ferramentas, o software em funcionamento mais que documentação abrangente, a colaboração com o cliente mais que negociação de contratos, e o responder a mudanças mais que seguir um plano. (MICHELS; FERREIRA, 2013).

Em outro estudo de revisão bibliográfica, elaborado por Mendonça e Alencar (2019), sobre metodologias ágeis subjacente ao custo de produção do software no setor público, verificou-se que após a identificação das configurações de métricas de tamanho de software e métodos ágeis mais aderentes ao setor público, evidenciou-se como plausíveis a métrica ponto de função (PF) e os métodos ágeis, em especial o Scrum.

Para o atendimento da necessidade de uma microempresa Trigo e Barreto (2019) recomendam como forma de suprir a necessidade de um levantamento de requisitos, bem como melhorar a aceitação por parte do cliente, o uso da metodologia ágil, em particular a metodologia XP, em face propiciar uma maior proximidade dos desenvolvedores e seus clientes.

Terlizzi e Biancolino (2014), propõe para o projeto de software no setor bancário o uso do método Scrum desde que sejam respondidas positivamente ao seguinte check-list: a equipe de desenvolvimento varia de 3 a 9 pessoas; é possível implantar funcionalidades relevantes em ambiente produtivo a cada 2-4 semanas; o escopo não está claramente definido; disponibilidade

de espaço físico para alocação da equipe; e os gerentes funcionais liberam seus colaboradores até a conclusão do projeto.

No cenário de uma pequena empresa de base tecnológica Carvalho e Mello (2012) propuseram uma pesquisa-ação utilizando a metodologia Ágil com uso do método Scrum, dividindo-a em fases a saber: fase exploratória - identificação da situação-problema; planejamento da implantação do Scrum; fase de ação - primeira iteração; fase de ação - segunda iteração; fase de ação - iteração final; e fase de avaliação - análise dos resultados.

Como prerrogativas de insucesso da aplicação do Scrum encontra-se a falta de conhecimento das estimativas de software, dificuldade do time no autogerenciamento, a falta de comprometimento do time, a influência e a importância da alta direção em todo o processo de implantação do método. (CARVALHO; MELLO, 2012).

Não obstante faz-se saber que ainda não sendo possível identificar o aumento da qualidade do produto, da satisfação de clientes e do retorno do investimento do projeto, o método Scrum foi condizente com a realidade da pequena empresa, pois se mostrou um processo focado em resultados, na comunicação da equipe e interação com os clientes sem desprezar as restrições enfrentadas. (CARVALHO; MELLO, 2012).

3 CONSIDERAÇÕES FINAIS

Pode-se afirmar que as metodologias tradicionais são consideradas pesadas e inflexíveis, principalmente porque logo no início do projeto o escopo se encontra fechado, assim qualquer alteração nos requisitos pode tornar o custo do software inaceitável, tornando tais metodologias obsoletas.

O manifesto Ágil surgiu como uma resposta aos problemas enfrentados pela engenharia de software, sobretudo as questões de alterações de requisitos e a complexidade dos novos sistemas, assim os métodos ágeis cada vez mais ganham adeptos no mercado tecnológico.

O grande diferencial das metodologias ágeis de desenvolvimento é o maior foco dado as pessoas e não aos processos, e os seus conjuntos de valores, princípios e práticas o que possibilita uma resposta mais rápida às constantes mudanças do mercado.

As aplicações das metodologias ágeis abrangem vários nichos de mercados, desde as micro e pequenas empresas até empresas públicas, por serem aderentes as características

individuais dessas organizações, estando sempre atentas ao escopo, ao prazo, ao custo, a natureza do processo e ao negócio da empresa, além dos seus objetivos estratégicos.

Conclui-se que as metodologias Ágeis provêm aos desenvolvedores de softwares o cumprimento de prazos e entregas softwares de qualidade, em vista que, tais métodos proporcionam pequenas iterações e entregas ao final de cada iteração, levando ao melhor aproveitamento dos recursos disponíveis e a garantia de entrega de um produto de qualidade que atenda as expectativas do cliente.

4 REFERÊNCIAS

ALLIANCE, A. **What is Extreme Programming (XP)**. 2018. Disponível em: <<https://www.agilealliance.org/glossary/xp/>>. Acesso em: 07/10/2020.

BESSA, Thiago; ARTHAUD, Daniel Dias Branco. **Metodologias ágeis para o desenvolvimento de softwares**. Ciência e Sustentabilidade – CeS. v. 4, n. 2, p. 173-213, jul./dez. 2018.

CARVALHO, Bernardo Vasconcelos de; MELLO, Carlos Henrique Pereira. **Aplicação do Método Ágil Scrum no Desenvolvimento de Produtos de Software em uma Pequena Empresa de Base Tecnológica**. Gest. Prod., São Carlos, v. 19, n. 3, p. 557-573, 2012.

CURADO, Luis Augusto Trindade; MACHADO, Giselle Barbosa Gomes; SILVA, Rogério Oliveira da. **Tecnologias em Projeção**, volume 7, número 2, ano 2016, página 55. Disponível em: <<http://revista.faculdadeprojecao.edu.br/index.php/Projecao4/article/download/693/618>> Acesso em: 22 out. 2020.

DENNIS, A.; WIXOM, B. H.; ROTH, R. M. **Análise e Projeto de Sistemas**. 5. ed. Rio de Janeiro: LTC, 2014.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010.

Manifesto para Desenvolvimento Ágil de Software. Disponível em: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>> Acesso em: 01 out. 2020.

MENDONÇA, Alexandre F.; ALENCAR, Fernanda M.R. **Métricas de Tamanho de Software com Métodos Ágeis no Setor Público: uma Revisão Sistemática**. Cadernos do IME: Série Informática: Vol. 42: Julho 2019.

MICHELS, Everton; FERREIRA, Marcelo Gitirana Gomes. **Gerenciamento Ágil no Processo de Desenvolvimento de Produtos Inovadores: Uma Análise Bibliográfica Sistemática**. Revista de Gestão e Projetos - GeP, São Paulo, v. 4, n. 1, p 52-76, jan./abr. 2013.

PRESSMAN, Roger S. **Engenharia de Software**. Mc Graw Hill, 6 ed, Porto Alegre, 2010.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software - Uma Abordagem Profissional**. 8. ed. São Paulo: AMGH Editora, 2016.

PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Porto Alegre: Editora Bookman, 2014. Disponível em: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>> Acesso em: 03 out. 2020.

SILVA, Todman Reis da; MEDEIROS, Marcus Vinicius Batella; MEDEIROS, Glaucia Rodrigues Nascimento. **Gestão de Riscos no Framework SCRUM Utilizando Análise SWOT**. Revista de Tecnologia da Informação e Comunicação da Faculdade Estácio do Pará. vol. 1, num. 1, pag. 48–5. Belém, Abril 2018. Disponível em: <<http://revistasfap.com/ojs3/index.php/tic/article/view/172>> Acesso em: 06 out. 2020.

SILVA, Daisy Eliana dos Santos; SOUZA, Ingredy Thaís de; CAMARGO, Talita. **Metodologias Ágeis para o Desenvolvimento de Software: Aplicação e o Uso da Metodologia Scrum em Contraste ao Modelo Tradicional de Gerenciamento de Projetos**. Revista Computação Aplicada, v. 2, n. 1, 2013. Disponível em: <<http://revistas.ung.br/index.php/computacaoaplicada/article/view/1408>> Acesso em: 01 out. 2020.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

SUTHERLAND, Jeff; SCHWABER, Ken. **The Scrum Guide™**. 2017. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>> Acesso em: 03 out. 2020.

TERLIZZI, Marco Alexandre; BIANCOLINO, César Augusto. **Projeto de Software no Setor Bancário: Scrum ou Modelo V**. TAC, Rio de Janeiro, v. 4, n. 1, art. 4, pp. 46-58, Jan./Jun. 2014.

TRIGO, Ramon Alves; BARRETO, Luiz Claudio. **Evolução dos Métodos de Desenvolvimento de Software em Microempresas**. Revista Gestão em Foco - Edição nº 11 – Ano: 2019. Disponível em: <http://portal.unisepe.com.br/unifia/wp-content/uploads/sites/10001/2019/08/009_EVOLU%C3%87%C3%83O-DOS-M%C3%89TODOS-DE-DESENVOLVIMENTO-DE-SOFTWARE-EM-MICROEMPRESAS.pdf> Acesso em: 06 out. 2020.

TSUI, F.; KARAM, O. **Fundamentos de Engenharia de Software**. Rio de Janeiro: LTC, 2013.

Adival José Reinert Junior

Graduado em Matemática (UNIASSELVI, 2010),
Especialista em Docência do Ensino Fundamental, Médio e Superior (FACISA/CELER, 2010),
Diretor Geral da Faculdade UniBF,
e-mail: contato@unibf.com.br.

Neide Pereira de Oliveira

Graduada em Sistemas de Informação (FAVI, 2002),

Especialista em Engenharia de Produção (IFES, 2013),
Especialista em Análise de Sistemas (UniBF, 2021)
Analista de Sistemas da Companhia Espiritossantense de Saneamento,
e-mail: neidenpo@gmail.com.

Recebido em 01/04/2021
Aprovado em 06/06/2021