Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

METODOLOGIAS ÁGEIS: um comparativo entre extreme programming (XP) e scrum

AGILE METHODOLOGIES: a comparison between extreme programming (XP) and scrum

Jean Carlos Albuquerque Souza[I]
Marcus Oliveira[II]

RESUMO

Com o mercado em constante crescimento, ocorreu uma evolução no âmbito de desenvolvimento de software, uma vez que os métodos antigos já não eram mais adequados às necessidades do cliente, fazendo surgir as metodologias de desenvolvimento ágil. Porém, com o surgimento de novos métodos ágeis, surgiram, também, dúvidas sobre qual metodologia pode ser a mais adequada para ser utilizada em um projeto. Diante disso, o presente artigo se propõe a avaliar duas das principais metodologias ágeis da atualidade, o Scrum e o Extreme Programming, com o fim de encontrar qual melhor se encaixe com a necessidade do desenvolvedor. Por meio de pesquisa bibliográfica, consulta de artigos, livros e sites, ambas as metodologias serão abordadas, por meio de um breve resumo de cada, bem como, serão apresentados alguns de seus pontos positivos e negativos. O objetivo disso, é auxiliar o desenvolvedor de software em uma futura tomada de decisão.

Palavras-chave: Metodologias Ágeis. Scrum. Extreme Programming.

ABSTRACT

With the market in constant growth, there was an evolution in the scope of software development, since the old methods were no longer adequate to the client's needs, giving rise to agile development methodologies. However, with the emergence of new agile methods, doubts also arose about which methodology might be the most suitable to be used in a project. Therefore, this article proposes to evaluate two of the main agile methodologies of today, Scrum and Extreme Programming, in order to find which, one best fits the developer's needs. Through bibliographical research, consultation of articles, books, and websites, both methodologies will be approached, through a brief summary of each, as well as some of their positive and negative points will be presented. The purpose of this is to assist the software developer in future decision making.

Keywords: Agile Methodologies. Scrum. Extreme Programming

Data de submissão do artigo: 17/11/2021.
Data de aprovação: 06/12/2021.
DOI: **10.52138/citec.v13i1.205**

[I] Estudante da Faculdade de Tecnologia (Fatec) –Taquaritinga – SP – Brasil. E-mail: jean.carlos.alb.souza@gmail.com
[II] Prof. Da Faculdade de Tecnologia (Fatec) –Taquaritinga – SP – Brasil. E-mail: marcus.oliveira10@fatec.sp.gov.br-Docente

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

1 INTRODUÇÃO

Segundo o Pressman (2011), os métodos de desenvolvimento ágeis se desenvolveram para sanar pontos fracos dos métodos de desenvolvimento convencionais. O mercado sempre está em constantes mudanças, as necessidades dos usuários também mudam conforme o tempo e novos competidores entram no mercado, por isso, é muito difícil que todos os requisitos de um projeto sejam definidos logo no início, portanto, é necessário ser ágil para se adaptar.

As metodologias ágeis estão cada vez mais ganhando espaço frente às metodologias tradicionais, visto que são mais flexíveis, menos burocráticas, mais dinâmicas e entregam resultados com mais rapidez (PASCUTTI *et al.,* 2019). Por isso, já se tornaram uma necessidade estratégica. Algumas das principais metodologias Ágeis usadas atualmente são o Scrum e o Extreme Programming (XP), cada uma com um foco diferente, sobre como se deve agir e operar em diferentes situações.

Segundo Sbrocco e Macedo (2012), O Scrum define uma abordagem para gerenciamento de projetos mais ampla, considerando um processo de desenvolvimento mais iterativo e incremental. Já o Extreme Programming (XP), tende a ser uma metodologia ágil com mais foco nas atividades de desenvolvimento e comunicação com o cliente. Com o grande crescimento no uso dessas metodologias em diferentes projetos, podem surgir dúvidas sobre qual delas é a mais adequada para um projeto, e uma escolha não acertada pode ocasionar diferentes tipos de problemas posteriormente.

O presente trabalho analisou os aspectos relevantes de cada uma dessas metodologias ágeis, sendo a Scrum e a Extreme Programming (XP), e realizar um comparativo entre às duas, para poder demonstrar qual método pode ser o mais compatível com a necessidade de diferentes desenvolvedores, em diferentes aspectos.

Esse artigo foi estruturado em quatro seções, sendo a primeira delas a introdução, trazendo referencial teórico de modo geral sobre às metodologias ágeis. Na seção dois é apresentada a fundamentação teórica, contendo as informações e característica mais detalhadas sobre o tema central da pesquisa. Na seção três é apresentado os comparativos entre. às duas metodologias. E por fim, na seção quatro, são apresentados os resultados obtidos com a pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Nessa seção são apresentados os conceitos referentes ao tema central da pesquisa deste artigo, o Scrum e Extreme Programming (XP), obtidos através da pesquisa bibliográfica em diferentes plataformas como artigos, monografias, livros e sites.

2.1 Metodologias ágeis

As metodologias ágeis constituem uma nova classe de metodologias de desenvolvimento de software, criada para atender à crescente pressão do mercado por processos mais ágeis e leves, com ciclos de desenvolvimento cada vez mais curtos (ABRAHAMSSON, 2003).

Com o passar do tempo, foi necessária uma evolução em relação à forma de gerenciar projetos, pois os modelos tradicionais como a cascata, por exemplo, não suportavam mais esse

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

processo, devido à falta de agilidade na entrega de software ao cliente. A partir dessa situação, foi preciso buscar melhores formas de gerenciamento (SOMMERVILLE, 2011).

Segundo Sbrocco e Macedo (2012), em 2001, uma importante reunião ocorrida em uma estação de esqui, marcou definitivamente o surgimento e a propagação de paradigmas de desenvolvimento do software ágil. Nesta reunião os profissionais de software que já trabalhavam com os chamados métodos leves utilizados na época, representavam diferentes metodologias, como SCRUM, Extreme Programming, DSDM, Adaptative Software Development, Crystal, Feature Driven Development, entre outras. O objetivo do encontro era trocar ideias sobre o que estavam fazendo e discutir formas de melhorar o desempenho de seus projetos. Cada participante trouxe suas experiências, teorias e práticas de como obter sucesso no desenvolvimento de projetos de software, portanto a aplicação de uma metodologia de desenvolvimento acabava diferindo das outras empresas. Durante a reunião, eles perceberam que apesar de cada um utilizar uma prática diferente ou adaptada à sua necessidade, todos acabaram concordando que existia um conjunto de princípios básicos que parecia ter sido respeitado por todos quando os projetos eram bem-sucedidos. E então eles criaram um documento que ficou conhecido mundialmente como manifesto ágil.

Das definições existentes de métodos ágeis, a mais aceita é: Métodos ágeis são um conjunto de práticas que seguem os princípios do Manifesto Ágil (BECK *et al.,* 2001)

2.2 Características do Scrum

O Scrum é uma metodologia de desenvolvimento ágil, criada por Jeff Sutherland em 1990 (PRESSMAN, 2006). O nome Scrum é surgiu de um estudo conduzido em 1986 por Takeuchi e Nonaka (1986) no artigo “The new product development game”, publicado na Harvard Business Review. Os autores perceberam que os times mais criativos, times que desempenhavam melhor que os demais, tinha uma certa liberdade para se gerenciar, ao comparar essas equipes de alto desempenho, percebeu que utilizavam bastante a formação "Scrum" existente nas equipes de rúgbi.

De acordo com o Sbrocco (2012), o uso dessa terminologia pareceu adequado, porque, no rúgbi, cada time age em conjunto, como uma unidade integrada, cada membro desempenha um papel específico e todos se ajudam em busca de um benefício em comum.

Segundo Sutherland (2016), o Scrum é um framework de gerenciamento de projetos, ele estimula as equipes a aprenderem com as experiências, a se organizarem enquanto solucionam algum problema e a refletirem sobre o sucesso e o fracasso, para sempre estarem evoluindo. Embora o Scrum seja uma das metodologias mais comum entre as equipes de desenvolvimento de software, os princípios e as lições que essa metodologia emprega, podem ser aplicados a todas as categorias de trabalho, que envolvam uma equipe. Esse é um dos motivos do Scrum ser umas das metodologias mais populares no mundo.

O objetivo do Scrum é entregar a maior qualidade de software possível dentro de uma série de pequenos intervalos de tempo fixo, chamados Sprints, que tipicamente duram menos de um mês (SUTHERLAND *et al.,* 2000).

Segundo Schwaber e Sutherland (2011), o Scrum é fundamentado nas teorias empíricas de controle de processo, ou empirismo. O empirismo afirma que o conhecimento vem da experiência e de tomada de decisões baseadas no que é conhecido. Ele se apoia em três pilares: transparência – Aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados; inspeção – Os usuários Scrum devem, frequentemente, inspecionar os artefatos Scrum e o progresso em direção ao objetivo, para detectar indesejáveis variações. A inspeção não pode sobrepor a execução das atividades; adaptação – Se for visto que algum aspecto do

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

processo se desviou dos limites aceitáveis e que o produto será inaceitável, o processo deve ser reajustado para produzir o que era esperado. E isso deve ser feito o mais breve possível para minimizar os desvios.

2.2.1 Vantagens do scrum

As vantagens do Scrun, de acordo com Sbrocco e Macedo (2012); Schwaber e Sutherland (2011); Sutherland (2016); Mango (2017), são:

- Adaptabilidade – O Scrum usa a imprevisibilidade que um grande projeto possui ao lidar com seus problemas de maneira eficiente. Com uma abordagem mais empírica e a entrega iterativa, fazem com que os projetos se tornem adaptáveis e abertos a mudanças durante seu desenvolvimento.
- Transparência – O Scrum observa e acompanhar todos os membros que fazem parte do projeto ou que fazem parte da organização. Facilitando conseguir se adaptar às mudanças e entender para qual caminho deve se seguir.
- Redução das falhas – Como a metodologia tem foco na qualidade faz com que haja uma redução dramática na quantidade de bugs nos softwares produzidos utilizando o Scrum.
- Melhoria Constante – Como o Scrum é divido por etapas, a cada nova etapa, pode se gerar um novo incremento e melhorar o que não ficou tão bom nas etapas anteriores. Gerando mais valor ao produto, pois, está sempre ajustando e melhorando durante o processo.

2.2.2 Desvantagens do scrum

As desvantagens do Scrun, de acordo com Sbrocco e Macedo (2012); Schwaber e Sutherland (2011); Sutherland (2016); Mango (2017), são:

- Visão segmentada – Combinando a uma visão segmentada que o Scrum tem, mais a tentativa de ser ágil pode levar a equipe a perder a visão do projeto na totalidade. Causando falhas críticas na hora de reunir as partes e concluir o programa.
- Falhas de documentação – Normalmente em projetos que utilizam o Scrum, o gerenciamento do projeto é dividido em etapas e, muitas vezes, apenas algumas etapas são documentadas corretamente. A falta de documentação sobre o andamento do projeto pode causar grandes problemas, em um futuro em que possam ser necessárias.
- Problemas com prazos – Como o Scrum tende a focar mais em qualidade do que no resultado, pode ser que os prazos não sejam estipulados adequadamente ao projeto, levando a um atraso do resultado.
- Desordem nas funções – Como o Scrum não tem a presença de papéis definidos, podem ocorrer alguns problemas relacionados a comunicação interna, deixando os programadores confusos quanto as suas tarefas.

2.3 Características do extreme programming (XP)

Segundo Wildt *et al.* (2015), O eXtreme Programming é uma metodologia ágil de desenvolvimento de software voltada para times de pequeno a médio porte, onde os requisitos são vagos e mudam frequentemente. O Extreme Programming é uma metodologia de desenvolvimento de software com foco em agilidade de equipes e na satisfação do cliente, apoiado em valores como simplicidade, comunicação, coragem, respeito e o feedback

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

constante. A comunicação é feita diretamente com o cliente; o feedback é dado a todo momento, para manter o projeto segundo o cliente; a simplicidade é percebida porque o XP foca em solucionar os problemas atuais do cliente, não se preocupando com os futuros e otimizando custos; respeito, pois, estimula o desenvolvimento de respeito mútuo entre cliente e desenvolvedores; coragem, porque a todo momento pode ter mudança no projeto e isso é um risco para os desenvolvedores que utilizam essa metodologia. Dessa forma, ela propicia que o projeto seja executado dentro do prazo e do orçamento, fazendo com que o cliente fique satisfeito e a equipe de desenvolvimento não tenha problemas devido a possíveis atrasos no projeto.

Segundo o Beck (2004), o XP é uma metodologia voltada para o desenvolvimento onde os requisitos podem se modificar constantemente com o andamento do projeto. O XP busca gerar o máximo de valor possível para o cliente, no menor espaço de tempo possível, além de gerar protótipos para que o cliente tenha a possibilidade de analisar e testar, para opinar e garantir que esteja como queria.

Segundo Sbrocco e Macedo (2012), o XP possui três características marcantes, que são, o feedback constante, a abordagem incremental e o encorajamento da comunicação entre as pessoas. Devido as essas características, o XP também ficou conhecido por ser a uma das metodologias de desenvolvimento de software menos formal. Além de ser uma metodologia de desenvolvimento de software que dá a preferência ao desenvolvimento orientado a objetos, sendo mais adequada para pequenas e médias equipes, além de se uma metodologia que utilizar o modelo incremental, ou seja, enquanto o software é utilizado, novas melhorias são implementadas. O XP sempre está buscando fundamentar suas práticas em um conjunto de valores bem definidos, além de se empenhar em assegurar que a equipe se concentre em desenvolver primeiramente o que é realmente necessário para o funcionamento do projeto, ao invés de colocar no projeto várias funcionalidades que o cliente não precisa no momento, e podem ser inseridas futuramente.

2.3.1 Vantagens Do Extreme Programming (XP)

As vantagens do extreme programming (XP), de acordo com Wildt *et al.* (2015); Sbrocco e Macedo (2012); Mango (2017), são:

- Planejamento – Trata-se de uma reunião realizada entre os clientes e desenvolvedores, feita no início do projeto ou de uma etapa dele, em que se é discutido o que priorizar e quais atividades focar. No XP o cliente deve estar presente acompanhando de perto o desenvolvimento do projeto, para poder ajudar a equipe a ter uma melhor compreensão das funcionalidades.
- Programação em Duplas – O XP tem a possibilidade de o desenvolvimento das atividades serem realizadas em duplas, onde o trabalho é feito com somente uma máquina onde um codifica, e o outro, crítica ou dá sugestões, conseguindo, dessa forma, otimizar a produção de código, e melhorando a qualidade.
- Design Simplificado – O código deverá estar em sua forma mais simples e clara possível, conforme os padrões definidos pelos membros da equipe de desenvolvimento, facilitando a visualização e a compreensão, tonando possível, caso necessário, a continuidade por qualquer membro da equipe.
- Desenvolvimento guiado por testes – Para todo código que será gerado durante o processo de desenvolvimento, deve existir um teste automatizado, que o verifique. Dessa forma, é possível garantir que todo código está funcionando de maneira correta.

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

2.3.2 Desvantagens do extreme programming (XP)

As desvantagens do extreme programming (XP), de acordo com Wildt et al. (2015); Sbrocco e Macedo (2012); Mango (2017), são:

- Estilo Consistente – As equipes envolvidas, os desenvolvedores e os clientes, podem divergir entre si, e internamente também, pelo fato de não possuírem total conhecimento da metodologia XP.
- Ausência do Cliente – O XP necessita ter o cliente sempre presente e acompanhando o projeto, se ele não estiver presente afeta SCHWABER o time desenvolvimento do projeto diretamente, pois, o cliente tem quer dar um feedback constante, conforme o andamento do projeto.
- Refactoring – É uma atividade que demanda um alto conhecimento e habilidade da metodologia XP, utilizar o refactoring pode acabar levando muito tempo e esforço para ser implementado.
- Ausência da avaliação de risco – O XP não nos oferece tanto suporte a análise de risco, como em outras metodologias, assim não é possível avaliar as estratégias capazes de diminuir os riscos e ameaças ao projeto.

3 QUADROS COMPARATIVOS ENTRE AS METODOLOGIAS

Após a apresentação sobre as metodologias Scrum e o Extreme Programming (XP), visto nos tópicos anteriores desse artigo, em que é apresentada de maneira geral sua história, como funciona, e algumas de suas vantagens e desvantagens, esse tópico servirá para retratar um comparativo entre algumas características e critérios comuns entre ambas, pretendendo prover mais aprofundamento e entendimento de cada metodologia e suas principais diferenças, para fornecer assistência em um processo de tomada de decisão.

Figura 1 – Comparativo entre ambas as metodologias

	Scrum	XP
Requisitos iniciais	Os requisitos são listados originando o Product Backlog	Histórias escritas pelo cliente
Atribuir requisitos aos incrementos	Os requisitos definidos no Product Backlog são alocados as Sprints durante a Reunião de Planejamento.	Definição das histórias que serão desenvolvidas a cada iteração
Projetar arquitetura do sistema	Projeto geral baseado no Product Log	Paralelo ao desenvolvimento das histórias
Desenvolver incremento de sistema	Implementação dos requisitos contemplados no Sprint Backlog.	Implementação das histórias que fazem parte da interação corrente por dupla de programadores

Fonte: Mango, 2017

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

Figura 2 – Comparativo entre ambas as metodologias

	Scrum	XP
Validar incremento	Acontece ao final da Sprint	Programados testam a unidade e cliente realiza a aceitação; os testes são escritos antes da codificação e então, executados
Integrar incremento	Acontece ao final de cada Sprint	Código é integrado a medida que vai sendo desenvolvido
Validar sistemas	Sistema é validado no último dia de cada Sprint	Sistema é validado pelo cliente
Entrega final	Todos os requisitos do Product Backlog desenvolvidos	Satisfação do cliente

Fonte: Mango, 2017

Também pode-se classificar se há a presença ou não de alguns critérios comum, em ambas as metodologias. Que será mostrado a seguir no quadro 3.

Figura 3 – Comparativo entre critérios presentes em ambas as Metodologias

	Scrum	XP
Reuniões diarias	X	
Planejamento iterativo	X	X
Quadro de tarefas	X	
tempo de ciclo	X	
Teste unitário		X
Participação do cliente	X	X

Fonte: Mango, 2017

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

4 CONCLUSÃO

Mediante o conhecimento proporcionado por meio das pesquisas bibliográficas e comparação entre as características das metodologias Scrum e o Extreme Programming (XP), foi possível sintetizar algumas considerações.

Pressupõem-se que não existe melhor ou pior entre as metodologias ágeis para um cenário genérico, e sim qual atende melhor às necessidades do desenvolvedor ou de um projeto. Desta forma, para compreender qual metodologia utilizar, faz-se necessário realizar o levantamento dos requisitos de cada metodologia junto ao cliente e, com base nesses requisitos, será possível indicar a metodologia que melhor atende às demandas, norteando-se por alguns aspectos, principalmente considerando as informações obtidas através deste artigo.

O Scrum é uma metodologia de gerenciamento mais amplo, que não se prende somente ao mundo de desenvolvimento de software. Sendo uma metodologia essencialmente interativa e incremental, ou seja, o programa é desenvolvido em etapas, e no fim de cada etapa uma nova se inicia. No Scrum, são aceitáveis mudanças durante o andamento do projeto, pois, estas mudanças podem ser implementadas em etapas futuras, sem influenciar diretamente a etapa atual do projeto. Durante cada etapa do projeto os desenvolvedores devem codificar e testar, para no fim entregar um protótipo funcional para que o cliente possa testar e avaliar.

Já metodologia XP tem sua aplicação mais flexível, mas que tem como foco no mundo de desenvolvimento de softwares. É uma metodologia possui valores bem marcantes como a comunicação, feedback, simplicidade, respeito e coragem. Com base nesses valores o XP consegue prover um ambiente agradável ao desenvolvedor. O XP foca em garantir que o projeto seja entregue dentro tempo e com o máximo de valor possível, visando sempre garantir que o cliente fique satisfeito com o produto.

Sendo assim, as lições aprendidas a partir desse estudo pretendem indicar meios para que os desenvolvedores de software possam tomar uma decisão de qual entre as metodologias propostas, deve utilizar mediante um cenário proposto, para que em breve não haja a necessidade de migração para outra metodologia em virtude da falta de conhecimento ou problemas técnicos.

REFERÊNCIAS

ABRAHAMSSON, P. *et al.* **New directions on agile methods: a comparative analysis. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING**, 25., 2003, Portland. Anais eletrônicos […]. Portland: ICSE, 2003.

BECK, K. **Programação extrema (XP) explicada**. [Porto Alegre]: Bookman. 2004.

BECK, K *et al*.. **Extreme programming explained: embrace change**. Boston: Addison-Wesley, 2001.

MANGO, Renan. **Comparativo entre as metodologias ágeis Scrum e XP para produção de softwares**. [*S. l.*]: Linkedin, 8 dez. 2017. Disponível em: https://www.linkedin.com/pulse/comparativo-entre-metodologias-%C3%A1geis-scrum-e-xp-para-produ%C3%A7%C3%A3o-mango/?originalSubdomain=pt. Acesso em: 16 nov. 2021.

PRESSMAN, Roger S. **Engenharia de Software**: Uma abordagem profissional. 7. ed. Porto Alegre: AMGH, 2011.

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

PASCUTTI, Márcia Cristina Dadalto *et al*. **Engenharia de Software**. Maringá-Pr.: Unicesumar, 2019.

PRESSMAN, Roger S. **Engenharia de Software**. 6. ed. São Paulo: McGraw-Hill, 2006.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar. **Metodologias Ágeis**: Engenharia de Software sob medida. São Paulo: Erica, 2012.

SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum**, 2011. Disponível em: http://www.scrum.org/storage/scrumguides/Scrum%20Guide%20%20Portuguese% 20BR. pdf. Acesso em: 06 dezembro. 2021.

SOMMERVILLE, Lan. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

SUTHERLAND, Jeff *et al* **Scrum**: a arte de fazer o dobro do trabalho na metade do tempo. Trad. Nina Lua. 2. ed. São Paulo: Leya, 2016. (E-book).

SUTHERLAND, J. *et al.* **Scrum:** an extension pattern language for hyper productive software development. [s.l.]: [s.e.], 2000.

TAKEUCHI, H; NONAKA, I.; **The new product development game**. Harvard Business Review, Cambridge, n. 1, jan. /feb., 1986.

WILDT, Daniel *et al*. **EXtreme Programming**: práticas para o dia a dia no Desenvolvimento Ágil de Software. São Paulo: Casa do Código, 2015. E-book.