

DOI: 10.5748/19CONTECSI/PSE/ESD/6982

**AN EXPLORATORY STUDY FOR THE CATALOGING OF AGILE PRACTICES
IN THE CONTEXT OF SOFTWARE DEVELOPMENT: A FOCUS ON THE
AGILE ALLIANCE**

**UM ESTUDO EXPLORATÓRIO PARA A CATALOGAÇÃO DE PRÁTICAS
ÁGEIS NO CONTEXTO DE DESENVOLVIMENTO DE SOFTWARE: UM FOCO
NO AGILE ALLIA**

Wilton Freitas Ribeiro ; <https://orcid.org/0000-0002-7689-8991>

UFPA - Universidade Federal do Pará

Sandro Ronaldo Bezerra Oliveira ; <https://orcid.org/0000-0002-8929-5145>

UFPA - Universidade Federal do Pará



AN EXPLORATORY STUDY FOR THE CATALOGING OF AGILE PRACTICES IN THE CONTEXT OF SOFTWARE DEVELOPMENT: A FOCUS ON THE AGILE ALLIANCE

ABSTRACT: Over the years software engineering specialists realized that the traditional development models no longer supported the constant changes in the projects requested either by the customers, or by external factors, for example, the change of technologies and Law Suit. Therefore, given these facts, they decided to form an alliance called Agile Alliance, and launched the agile manifesto with its values and principles, where the main focus was the interaction between people, working products, collaboration with customers and quick responses to changes. After the release of the manifest, numerous agile practices were suggested in order to improve the software development process, making them faster and more efficient, reducing the waiting time between the release of increments and adding more value to the final product, increasing customer satisfaction. customers. In this article, a catalog of the main agile practices is presented, listed by the Agile Alliance organization, in the agile subway map, the main objective of cataloging is to provide a collection of practices that can be used in different software development projects, in addition to informing its main features and benefits and propose a model of how to use a particular practice. A bibliographic research was carried out, in order to find studies in the literature that speak of the approached practice, in this way it is possible to suggest the complement of the studies from the works also cataloged along with the agile practices.

Keywords: Agile Methods, Agile Practices, Agile Alliance.

UM ESTUDO EXPLORATÓRIO PARA A CATALOGAÇÃO DE PRÁTICAS ÁGEIS NO CONTEXTO DE DESENVOLVIMENTO DE SOFTWARE: UM FOCO NO AGILE ALLIANCE

RESUMO: Ao longo dos anos especialistas em engenharia de software perceberam que os modelos de desenvolvimento tradicionais já não comportavam as constantes mudanças nos projetos solicitados tanto pelos clientes, quanto por fatores externos, por exemplo, a mudança de tecnologias e processos. Diante destes fatos, decidiram formar uma aliança denominada como Agile Alliance, e lançaram o manifesto ágil, com seus valores e princípios, onde o foco principal é a interação entre pessoas, produtos funcionando, colaboração com os clientes e rápida respostas as mudanças. Após o lançamento do manifesto, inúmeras práticas ágeis sugeriram afim de melhorar o processo de desenvolvimento de software, tornando-os mais rápidos e eficientes, diminuindo o tempo de espera entre o lançamento dos incrementos e agregando mais valor ao produto final, aumentado a satisfação dos clientes. Neste artigo, é apresentado um catálogo das principais práticas ágeis, elencadas pela organização Agile Alliance, no mapa do metrô ágil, o principal objetivo da catalogação é disponibilizar um acervo de práticas que poderão ser utilizadas em diferentes projetos de desenvolvimento de software, além de informar suas principais características e benefícios e propor um modelo de como utilizar determinada prática. Foi feita uma pesquisa bibliográfica, afim de encontrar estudos na literatura que falam da prática abordada, dessa forma é possível sugerir o complemento dos estudos a partir dos trabalhos também catalogados junto com as práticas ágeis.

Palavras-chave: Métodos Ágeis, Práticas Ágeis, Agile Alliance.

Agradecimentos: Este trabalho pertence ao projeto SPIDER/UFPA (<http://www.spider.ufpa.br>).

1. INTRODUÇÃO

As metodologias ágeis de desenvolvimento de software são consideradas como um conjunto de práticas e métodos, que tem como objetivo principal tornar o processo de desenvolvimento de software mais rápido, com menor custo e maior qualidade (FOWLER, 2005). O termo “Metodologia Ágil” tornou-se conhecido no ano de 2001, depois que um grupo de especialistas em processo de desenvolvimento de software representando diferentes métodos ágeis se reuniram para discutir maneiras de melhorar o desempenho dos projetos de software (PONTES E ARTHAUD, 2018).

Durante a reunião perceberam que em seus projetos anteriores havia um conjunto de práticas comum que, quando utilizadas, os projetos em sua grande maioria davam certo. Após essa reunião surgiu a aliança ágil e criou-se o Manifesto ágil com suas filosofias, valores e princípios (PONTES E ARTHAUD, 2018). Segundo Pressman (2011) as metodologias ágeis foram criadas para corrigir fraquezas reais e perceptíveis da engenharia de software tradicional. De acordo com Conceição e Silveira (2015) o desenvolvimento ágil oferece benefícios importantes, no entanto, o mesmo não é indicado para todos os projetos e produtos. Portanto deverá ser realizado uma análise antes do seu uso.

No Brasil o termo “Metodologia Ágil” vem se popularizando principalmente nos últimos anos, porém o que poucos imaginam é que devido a utilização de uma abordagem simplificada nos processos, essa metodologia habitualmente acaba sendo erroneamente confundida como um método com falta de controle e completa bagunça nos processos (MACHADO E MEDINA, 2009). No entanto, atualmente ser ágil é fazer a diferença em relação aos concorrentes e diferente do que muitos imaginam exigem muita organização e disciplina nos processos. (MACHADO E MEDINA, 2009).

No âmbito da Tecnologia da Informação existe uma cobrança excessiva para que os projetos obtenham resultados satisfatório, com entregas cada vez mais rápidas e custos reduzidos. Empresas buscam de todas as formas manter-se competitivas no mercado, e os profissionais de TI lutam para acompanhar o ritmo dessas mudanças. (MACHADO E MEDINA, 2009). Portanto existe um crescimento exponencial no desenvolvimento de metodologias ágeis, cujo o principal objetivo é aumentar a produtividade do trabalho com o compromisso de flexibilidade, qualidade e agilidade (MACHADO E MEDINA, 2009).

Estudos apontam que a utilização dessas práticas aumenta os índices de sucesso nos projetos de desenvolvimento de software, e evidenciam a importância em propagar o conhecimento das metodologias ágeis. (SCHOEFFEL, 2021). Ainda de acordo com estudo embora as práticas ágeis sejam muito utilizadas atualmente, existe uma ausência de referências bibliográficas quando comparadas com as metodologias tradicionais. E um dos fatores do fracasso nos projetos que adotam esta metodologia está justamente no baixo conhecimento e experiência na utilização dessas práticas. (SCHOEFFEL, 2021).

Portanto, este trabalho tem por objetivo apresentar um catálogo com as práticas ágeis elencadas no Mapa do Metrô Ágil do Agile Alliance, expondo suas principais características, além de sugerir como realizar tais práticas. Visto que durante o processo de desenvolvimento de software, geralmente a equipe de desenvolvimento se depara com problemas que habitualmente impedem a entrega de projetos dentro do prazo e do orçamento estipulado, ou em casos extremos com qualidade inferior ao que foi previsto (PONTES E ARTHAUD, 2018). Nesse contexto, surge a importância da divulgação das práticas ágeis.

A catalogação das práticas ágeis aqui apresentada, justifica-se como uma forma de contribuir com as discussões acerca das diferentes formas de utilização dessa metodologia, e conduzirão a melhores práticas de desenvolvimento de software. O artigo está organizado em seções. Na seção 2 é apresentado a metodologia da pesquisa. Na seção 3 é descrita a catalogação das práticas ágeis utilizada na realização deste trabalho de pesquisa. Na seção 4 apresenta a conclusão do trabalho, relatando as contribuições, limitações e trabalho futuros, e

por fim são apresentadas as referências bibliográficas utilizadas no desenvolvimento deste trabalho.

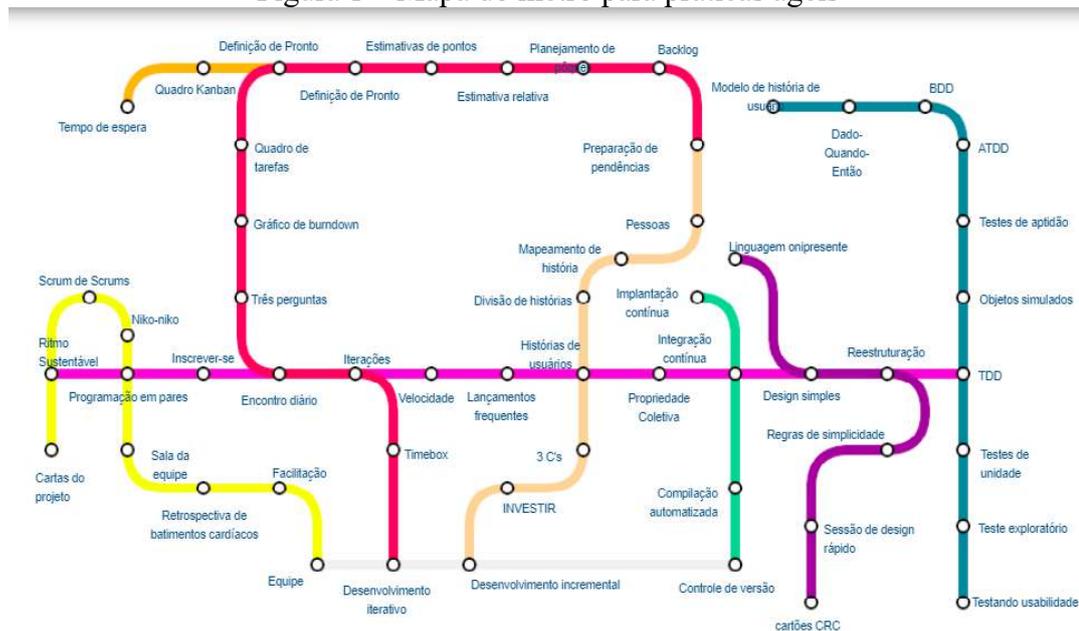
2. METODOLOGIA DA PESQUISA

Esta seção apresenta os métodos de pesquisa utilizados para a condução da catalogação das práticas ágeis. Com intuito de tornar os resultados mais confiáveis, reproduzíveis e auditáveis foram descritos os principais procedimentos utilizados para a realização deste trabalho.

A classificação das pesquisas quanto aos seus objetivos gerais pode ser classificada como exploratórias. Este tipo de pesquisas tem como principais objetivos explorar, estudar, analisar um determinado assunto para gerar um maior entendimento do problema (GIL, 2002). E também pode ser considerada como bibliográfica pois foi realizada buscas em jornais, revistas, livros, artigos científicos e materiais que já publicaram algo sobre o tema da pesquisa (GIL, 2002). Com o propósito de identificar trabalhos relacionados a utilização de práticas ágeis no processo de desenvolvimento de software. Foi priorizado as análises das definições de cada prática ágil, bem como suas principais características e benefícios.

Para a delimitação da pesquisa foram selecionadas apenas as práticas elencadas no Mapa do Metrô para Práticas Ágeis, disponibiliza pela organização Agile Alliance, que é uma comunidade sem fins lucrativos, que apoia organizações e pessoas que exploram, aplicam e expandem princípios e valores das práticas ágeis (AGILE ALLIANCE, 2022). A figura abaixo exibe o mapa do metrô utilizado como base para elaboração da catalogação das práticas ágeis.

Figura 1 – Mapa do metrô para práticas ágeis



Fonte: Agile Alliance (2022)

3. CATALOGAÇÃO DAS PRÁTICAS ÁGEIS

Esta seção apresenta a catalogação das práticas ágeis listada pela organização Agile Alliance.

3.1. Ritmo Sustentável

Essa prática recomenda que sejam realizadas no máximo oito horas diárias de trabalho e que preferencialmente não sejam realizadas horas extras. A fim de que os envolvidos não se sintam sobrecarregados e diminuam a concentração e disposição no processo de criação (AGILE ALLIANCE, 2022).

Principais características e benefícios: Pode-se afirmar que o principal objetivo desta prática é manter um trabalho sustentável e saudável, ou seja, um ambiente harmonioso, o que resulta em equipes com maiores índices de concentração, rendimento e produtividade (TELES, 2006).

Presente nas metodologias ágeis: Extreme Programming e Teams.

Estudo na literatura que menciona a prática abordada: A implantação das metodologias ágeis de desenvolvimento de software Scrum e Extreme Programming (XP): Uma alternativa para pequenas empresas do setor de tecnologia da informação, 2016.

Modelo para Utilização da Prática: Pensando na saúde e no bem-estar dos membros da equipe, o planejamento da quantidade de horas de trabalho deverá ser definido criteriosamente e realisticamente. Afim de evitar sobrecargas no trabalho diário, além de proporcionar um ritmo saudável, contribuído de forma positiva para o êxito dos projetos (BASSI FILHO, 2008).

3.2. Programação em Par:

Prática em que dois programadores trabalham lado a lado em um computador, cooperando entre si de forma contínua em um mesmo projeto. Durante o desenvolvimento um programador realiza a codificação enquanto o outro realiza a inspeção imediata de todo o código que é produzido (WILLIAMS E KESSLER, 2003).

Principais características e benefícios: Um dos principais objetivos desta prática é proporcionar o trabalho colaborativo entre os programadores, onde os mesmos poderão analisar em conjunto todos os detalhes do código, trocando aprendizados entre si, e principalmente evitando que erros ocorram, visto que caso alguma inconsistência seja encontrada um dos programadores já detecta e o problema logo é corrigido. (BASSI FILHO, 2008).

Presente nas metodologias ágeis: Extreme Programming e Teams.

Estudo na literatura que menciona a prática abordada: Um estudo de caso da adoção das práticas e valores do extreme programming, 2005.

Modelo para Utilização da Prática: Para a realização desta prática basicamente dois programadores deverão trabalhar lado a lado em um único computador, alternando entre os papéis de condutor (quem realiza a digitação/codificação) e o navegador que realiza a análise do código identificando possíveis erros e sugerindo melhorias (BASSI FILHO, 2008)

3.3. Inscreva-se para Tarefas

Prática em que os membros de uma determinada equipe de desenvolvimento frequentemente escolhem em quais tarefas trabalhar, em vez de receber o trabalho atribuído por terceiros. A tarefa escolhida poderá ser negociada com os demais membros da equipe. Tais discussões são realizadas habitualmente em pé diante do quadro de tarefas, geralmente durante a reunião diária (AGILE ALLIANCE, 2022).

Principais características e benefícios: Destaca-se nesta prática a possibilidade da atribuição de tarefas por meio da auto-seleção ao invés da alocação ser realizada por outra pessoas, por exemplo Coordenadores ou Gerentes de Projeto (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Extreme Programming

Estudo na literatura que menciona a prática abordada: Não foram encontrados na literatura documentos que relatam ou mencionam a utilização desta prática.

Modelo para Utilização da Prática: Para a realização desta prática é necessário a escolha de uma determinada tarefa que geralmente está representada por uma anotação no cartão de índice ou nota adesiva (AGILE ALLIANCE, 2022).

3.4. Daily Meeting

É uma reunião diária, de no máximo quinze minutos, realizada geralmente pela parte da manhã, para analisar o que foi feito, qual a estratégia a ser adotada para as próximas vinte quatro horas e o que está impedindo o time de avançar. Esta prática é também conhecida por outros nomes entre os usuários de metodologias ágeis, por exemplo, Daily Scrum, Daily Meeting, Stand-Up Meeting, Daily Stand-Up ou até mesmo Scrum Diário (AGILE ALLIANCE, 2022).

Principais características e benefícios: Um dos maiores benefícios desta prática é estabelecer as prioridades a serem realizadas nas próximas 24 horas, ou seja, o time elabora uma estratégia para tentar cumprir as atividades e/ou solucionar os possíveis problemas que estão pendentes. Dando maior agilidade e fluidez nos processos e contribuindo com a comunicação da equipe (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Extreme Programming e Scrum.

Estudo na literatura que menciona a prática abordada: SCRUM – Método Ágil: uma mudança cultural na Gestão de Projetos de Desenvolvimento de Software, 2017.

Modelo para Utilização da Prática Deverá ser realizado reunião preferencialmente pela parte da manhã e com duração de no máximo 15 minutos, com principal objetivo de todo o time responder as seguintes perguntas: “O que foi feito para o projeto desde a última reunião?”, “O que será feito para o projeto até a próxima reunião?”, “Existe algum impedimento para conseguir seu objetivo?” (MACHADO, MEDINA, 2009).

3.5. Iterações

Esta prática ágil se refere a um timebox no período que ocorre o desenvolvimento, sua duração poderá variar entre uma a quatro semana. O período de duração é geralmente é fixado no início dos projetos e poderá variar de um projeto para outro (MATTIOLI et al, 2009). Na metodologia SCRUM as iterações são também conhecidas como “Sprints” e na maioria das vezes sua duração é de 30 dias. Já no contexto XP as interações são conhecidas como “ciclos semanais” (AGILE ALLIANCE, 2022).

Principais características e benefícios: O acompanhamento na evolução dos projetos no contexto interativo é um dos principais benefícios que esta prática proporciona, pois é possível analisar se em cada iteração estão sendo atendidos os critérios de desenvolvimento e se a sua evolução está dentro do proposto, facilitando o acompanhamento pelos líderes dos projetos (MATTIOLI et al, 2009).

Presente nas metodologias ágeis: Extreme Programming e Scrum.

Estudo na literatura que menciona a prática abordada: Uma Proposta para o Desenvolvimento Ágil de Ambientes Virtuais, 2009.

Modelo para Utilização da Prática: Para a utilização desta prática deverá ser escolhido primeiramente em que contexto a mesma será utilizada, visto que dentro das metodologias Scrum ou XP seu entendimento relacionado a duração poderá variar,

posteriormente, os itens da iteração deverá ser definido assim como a duração e o responsável pelo acompanhamento da evolução, essa prática geralmente é realizada pelo gerente de projetos.

3.6. Velocidade

A prática visa mensurar a quantidade de trabalho que a equipe entrega em cada iteração. (BECK & FOWLER, 2001). Além disso, ao final de cada iteração, a equipe soma as previsões de esforço aliadas às histórias de usuários que foram concluídas com sucesso durante essa iteração, esse total de atividade realizadas são chamadas de velocidade de entrega. (AGILE ALLIANCE, 2022).

Principais características e benefícios: Com o decorrer dos anos esta prática tem sido muito valorizada, pois antes a grande preocupação dos projetos era a de não cometer erros, deixando a velocidade em segundo plano, porém este cenário mudou visto que o desenvolvimento rápido do software tem diversas vantagens, por exemplo, atender as necessidades de clientes que precisam de entregas cada vez mais rápidas devido a necessidade de seu tipo de negócio, que não podem esperar muitos meses de desenvolvimento (FADEL, SILVEIRA, 2010).

Presente nas metodologias ágeis: Extreme Programming.

Estudo na literatura que menciona a prática abordada: Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum e Lean, 2010.

Modelo para Utilização da Prática: Esta prática recomenda que sejam contabilizadas quantas tarefas foram concluídas dentro do prazo determinado, por meio deste cálculo o time poderá realizar uma estimativa da velocidade, além de verificar quanto tempo precisa para a conclusão de um determinado projeto.

3.7. Lançamento Frequentes:

Essa prática orienta que sejam realizados pequenos lançamentos de forma contínua e esses lançamentos devem ter valor de negócio para o cliente (BECK, 2004).

Principais características e benefícios: Com a prática de lançamentos frequentes é possível realizar a redução de riscos, pois essa prática mitiga se existiu falhas de planejamento, pois possibilita a validação com o cliente muito mais rápido, ou seja, existe um feedback do que deu certo e o que precisa ser corrigido (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Extreme Programming.

Estudo na literatura que menciona a prática abordada: Definição de um método que estabelece critérios para priorização de novos projetos e aplicação em um processo de desenvolvimento de software, 2013.

Modelo para Utilização da Prática: O lançamento frequente poderá ser realizado desde que todo o planejamento necessário tenha sido alinhado com todos os membros da equipe, definindo os critérios de cada entrega tal alinhamento também deve ser feito junto ao cliente.

3.8. Histórias de Usuários

Prática ágil para a representação de requisitos de software na forma de histórias de usuários, ou seja, curtas descrições de funcionalidades de software contadas a partir da perspectiva de um usuário (LONGO e SILVA, 2014).

Principais características e benefícios: A mudança de ideia sobre os detalhes ou a prioridade de agendamento das histórias de usuário que ainda não foram implementadas pelos desenvolvedores e que não interfira diretamente no andamento do projeto é um

dos principais benefícios dessa prática. Além de propor feedbacks contínuos, potencializando as competências e criatividade de cada um (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Extreme Programming e Product Management.

Estudo na literatura que menciona a prática abordada: A Utilização de Histórias de Usuários no Levantamento de Requisitos Ágeis, 2014.

Modelo para Utilização da Prática: Para a elaboração da história de usuários alguns critérios deverão ser seguidos, por exemplo, o número sequencial para a identificação dos cartões, o título e a descrição da história do usuário e a definição da prioridade de cada história. De acordo com a necessidade do projeto os cartões poderão ser alterados, desde que não interfira em itens já desenvolvidos (LONGO e SILVA, 2014).

3.9. Propriedade Coletiva

Prática que enfatiza a colaboração no código por todos os membros da equipe de desenvolvimento, ou seja, não existe o conceito de propriedade exclusiva ou restrita do código. Se um dos programadores identificar um erro ou uma possível melhoria no código, poderá realizar a alteração no mesmo sem nenhum impedimento, desde que realize anteriormente os testes e repasse para demais membros da equipe a mudança a ser realizada (BASSI FILHO, 2008).

Principais características e benefícios: Uma das principais características dessa prática está na possibilidade de todos os membros da equipe fazer alterações no arquivo de código conforme necessário: seja para concluir uma tarefa de desenvolvimento, para corrigir um defeito ou mesmo para realizar uma melhoria na estrutura geral do código, entre os maiores benefícios pode-se citar a validação constante do código por diversos programadores, ou seja, erros são minimizados pois a análise do código é constante e a identificação e correção de erros muito mais rápidas (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Extreme Programming.

Estudo na literatura que menciona a prática abordada: Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software, 2004.

Modelo para Utilização da Prática: Esta prática requer que um sendo de propriedade coletiva de código, onde todos os membros da equipe de desenvolvimento possam contribuir com o desenvolvimento, ajustes e melhorias nos códigos. Portanto não existe um único proprietário ou responsável pelo código e sim uma responsabilidade coletiva (BASSI FILHO, 2008).

3.10. Integração Contínua

Prática em que os membros da equipe frequentemente integram seus trabalhos, em cada integração são realizados teste para verificar se houve algum problema durante ou após a integração da nova funcionalidade. Após o sucesso na integração o sistema passa a contar com nova versão funcional do projeto a ser demonstrado e utilizado por toda a equipe (BECK, 2004).

Principais características e benefícios: A prática busca minimizar a duração e os esforços exigidos para a realizar as integrações, buscam também entregar uma versão do produto otimizada para lançamento a qualquer momento. Portanto requer um procedimento de integração reprodutível e automatizado, utilizando ferramentas disponíveis que auxiliam neste processo (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Extreme Programming e Devops.

Estudo na literatura que menciona a prática abordada: Análise comparativa das metodologias ágeis: SCRUM, XP E FDD, 2017.

Modelo para Utilização da Prática: Para a realização dessa prática deverá ser realizada a Integração contínua de software.

3.11. Designer Simple

A prática se refere a criação de códigos mais simples e de fácil compressão, mas que atenda todas as necessidades do projeto (BECK, 2000).

Principais características e benefícios: a prática tem como principal característica a utilização do conceito da sigla YAGNI, traduzindo literalmente “Você não vai precisar disso”; essa técnica é utilizada quando um programador tenta propor algo que não precisa ser implementado naquele momento, ou seja, não trará benefícios ao projeto (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Extreme Programming e Design.

Estudo na literatura que menciona a prática abordada: Extreme programming: metodologia ágil, 2013.

Modelo para Utilização da Prática: Para utilizar essa prática o time deverá desenvolver um design simples para seus códigos, afim de manter um bom entendimento entre todos os membros da equipe, sem a inclusão de trechos de códigos de forma desnecessária, ou seja, e dever de todos manter um código simples e eficiente.

3.12. Reestruturação (Refatoração)

Essa prática busca otimizar internamente a estrutura do código fonte de um software existente, mantendo o seu comportamento externo. Trata-se de uma determinada transformação de preservação de comportamento, por exemplo o fato de “Extrair Método” ou “Introduzir Parâmetro” (AGILE ALLIANCE, 2022).

Principais características e benefícios: O maior benefício desta prática está no processo realizar mudanças em um código funcional sem alterações em seu comportamento externo. Aprimorando estruturas internas afim de otimiza-la e prevenir de possíveis bugs (ASTELS, 2003).

Presente nas metodologias ágeis: Extreme Programming e Design.

Estudo na literatura que menciona a prática abordada: Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum e Lean, 2010.

Modelo para Utilização da Prática: Para a realização desta pratica os programadores deverão ter acesso aos códigos fontes, posteriormente deverão realizar a otimização dos códigos fontes sem afetar o funcionamento do mesmo.

3.13. Test Driven Development (TDD).

A prática consiste na realização de pequenas iterações onde novos casos de teste são escritos contemplando uma nova funcionalidade ou melhoria, posteriormente existe a necessidade do código passar por testes até ser implementado (FEITOSA, 2007). Em sua tradução literal significa desenvolvimento orientado a testes, essa prática também é conhecida como Test First Design (TFD), Test First Programming (TFP) (GEORGE; WILLIAMS, 2003).

Principais características e benefícios: Um dos maiores benefícios dessa prática consiste em reduções significativas nas taxas de defeitos (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Extreme Programming e Testing.

Estudo na literatura que menciona a prática abordada: Análise de Metodologias Ágeis: Conceitos, Aplicações e Relatos sobre XP e Scrum, 2011.

Modelo para Utilização da Prática: Para a realização dessa prática deverão ser seguidos os seguintes passos: adicione um pequeno teste; rode todos os testes e verifique as falhas que aparecerem; faça pequenos ajustes; realize novamente os testes e seja bem-sucedido; caso existam duplicações faça a remoção por meio do processo de refatoração (BECK, 2000).

3.14. Equipe

Essa prática ágil considera como “equipe” um grupo de pessoas atribuídas ao mesmo projeto ou atividade por um determinado período. Esta equipe tem responsabilidades compartilhadas e os resultados gerados sejam eles bons ou ruins serão atribuídos a toda equipe (AGILE ALLIANCE, 2022).

Principais características e benefícios: Inúmeros benefícios no trabalho em equipe, entre eles a comunicação, o compartilhamento de conhecimento entre os membros da equipe, decisões compartilhadas e um objetivo comum que é o sucesso do projeto (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Teams e Fundamentals.

Estudo na literatura que menciona a prática abordada: Uso eficaz de métricas em métodos ágeis de desenvolvimento de software, 2007.

Modelo para Utilização da Prática: Para a realização dessa prática um grupo de pessoas deverá trabalhar em conjunto para atingir um determinado objetivo.

3.15. Facilitação

Nessa prática uma pessoa tem a responsabilidade de conduzir a reunião, realizando o papel de facilitador, dando subsídios para que os objetivos das reuniões sejam atingidos. Porém não deverá entrar em discursões sobre o tema da reunião, apenas focar no seu andamento (AGILE ALLIANCE, 2022).

Principais características e benefícios: Essa prática tem como principal característica a transformação das reuniões, pois o facilitador fará o que for possível para que tudo ocorra da melhor forma possível e que os objetivos das reuniões sejam alcançados (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Teams.

Estudo na literatura que menciona a prática abordada: Fatores humanos na dependabilidade de sistemas de softwares desenvolvidos com práticas ágeis, 2014.

Modelo para Utilização da Prática: Para a realização dessa prática deverá ser nomeado um facilitador para as reuniões, e o mesmo deverá conduzir a reunião para que todos os temas sejam concluídos.

3.16. Retrospectiva

Prática ágil em que a equipe se reúne regularmente para refletir sobre cada detalhe dos eventos mais significativos ocorridos desde a reunião anterior e tomar decisões visando a correção de problemas ou a melhoria dos processos, tal prática também é conhecida como: “sprint retrospective”, “iteration retrospective” ou “sprint retro” (AGILE ALLIANCE, 2022).

Principais características e benefícios: Uma das principais características da sprint retro está na possibilidade de identificar e priorizar o que poderá ser melhorado e aplicado na próxima Sprint (SCHWABER e SUTHERLAND, 2009).

Presente nas metodologias ágeis: Teams.

Estudo na literatura que menciona a prática abordada: O impacto do uso das metodologias ágeis scrum e xp na satisfação dos stakeholders, 2014.

Modelo para Utilização da Prática: Para a realização dessa prática a equipe deverá ser reunir ao final da Sprint e realizar alguns questionamentos, por exemplo, “Quais os itens foram bem-sucedidos?”, “Quais as possíveis melhorias a ser implementada?” e “Quais os itens são prioridades?”. De acordo com as respostas destas perguntas a próxima Sprint será otimizada

3.17. Sala de Equipe

Prática que sugere que todos estejam em um mesmo local e se possível um representante do cliente. O ambiente deverá contar com ferramentas para o desenvolvimento das atividades,

por exemplo, estações de trabalho, quadros brancos e flipcharts. Além de espaço na parede para expor quadros de tarefas, planos de projeto ou outros gráficos. (AGILE ALLIANCE, 2022).

Principais características e benefícios: A comunicação e o ponto forte das salas de equipe, visto que todos estão em um ambiente propício em busca de um objetivo comum, além de contar com várias ferramentas para melhoria de performance da equipe (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Teams.

Estudo na literatura que menciona a prática abordada: Não foram encontrados na literatura documentos que relatam ou mencionam a utilização desta prática.

Modelo para Utilização da Prática: Para a realização dessa prática uma sala deverá ser preparada com os equipamentos e mobílias necessárias para a fluidez do trabalho em equipe.

3.18. Calendário Niko-Niko

Essa prática recomenda que seja instalado um calendário Niko-Niko em uma parede da sala da equipe. Esse calendário permite que cada membro da equipe registre uma avaliação gráfica de seu humor durante aquele dia, ao final de cada jornada de trabalho. (“Niko” significa “sorriso” na língua japonesa) (AGILE ALLIANCE, 2022).

Principais características e benefícios: Verificar como está a motivação e o bem-estar dos membros da equipe e o principal objetivo dessa prática, além da possibilidade acompanhar a evolução do humor da equipe, através das reações expostas no calendário, tal medida permite que os gestores ou líderes adotem práticas para manter a equipe motivada e com bom humor, visto que neste cenário tendem a produzir mais (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Teams.

Estudo na literatura que menciona a prática abordada: Métricas de avaliação para abordagens ágeis em projetos de software, 2014.

Modelo para Utilização da Prática: Para a realização dessa prática os membros deverão registrar suas emoções no calendário Niko-Niko, poderão ser utilizados emoticon coloridos, obedecendo a um pequeno código de cores, por exemplo, azul para um dia ruim, vermelho para neutro, amarelo para um bom dia. (AGILE ALLIANCE, 2022).

3.19. Scrum de Scrums

Prática que em que um membro da equipe é escolhido para participar de uma nova reunião com outros membros escolhidos por outras equipes (APPELO, 2010). Nessa reunião serão discutidos temas de interesses de todas as equipes, e o objetivo comum e retirar impedimentos que possam surgir no projeto entre as equipes e buscar melhor fluidez para os processos. Essa prática também é conhecida como Meta Scrum (AGILE ALLIANCE, 2022).

Principais características e benefícios: Um dos benefícios do Scrum de Scrum é a possibilidade rastrear os itens que estão causando impactos no andamento do projeto, como os representantes de todas as áreas estarão presentes, muitos itens acabam sendo solucionados na própria reunião. Além disso, poderá ser criado durante as reuniões metas a serem cumpridas por todas as equipes até a próxima reunião (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Teams.

Estudo na literatura que menciona a prática abordada: Implantação de métodos ágeis em empresa de desenvolvimento de software: um estudo de caso, 2015.

Modelo para Utilização da Prática: Para realização dessa prática deverá ser nomeado um representante da equipe, o mesmo deverá expor na reunião de scrum de scrum todos os itens que estão impedindo os avanços dos trabalhos relacionados a dependências a outras equipes, além disso, deverá responder sobre questionamentos acerca do trabalho de sua equipe.

3.20. Cartas de Projetos:

Essa prática orienta que seja criado um documento de alto nível dos principais fatores de sucesso do projeto, o mesmo deverá ser resumido o suficiente para que possa ser exibido em uma parede da sala da equipe como uma folha de papel do tamanho de um flipchart. No documento os itens que não podem faltar são: principais objetivos do projeto, limites de escopo e as partes interessadas (AGILE ALLIANCE, 2022).

Principais características e benefícios: Uma das características dessa prática está em deixar claro os objetivos e o andamento do projeto, assim como outros interesses comuns da equipe referente ao projeto, ou seja, o documento alinha de forma clara e objetiva os desafios em comum da equipe.

Presente nas metodologias ágeis: Teams.

Estudo na literatura que menciona a prática abordada: Não foram encontrados na literatura documentos que relatam ou mencionam a utilização desta prática.

Modelo para Utilização da Prática: Para a utilização dessa prática a equipe deverá descrever de forma clara e objetiva todos os principais itens do projeto, após a análise do documento por todos os membros da equipe, o mesmo deverá ser fixado na parede.

3.21. Quadro Kanban

Essa prática ágil visa a organização, otimização e produtividade dos projetos, geralmente sendo composta por quatro etapas (Colunas): To do (a fazer), Doing (Fazendo), Testing (Testando) e Done (feito). Os membros da equipe colocam no quadro Kanban as tarefas que precisam ser realizadas e à medida que vão sendo feitas elas mudam de coluna. (ARBULU; BALLARD; HARPER, 2003)

Principais características e benefícios: A utilização dessa prática possui inúmeros benefícios entre eles estão a melhoria na visualização do fluxo de trabalho, definição de prioridades e foco na melhoria contínua. Além de possibilitar que toda a equipe faça a análise e o controle do andamento de suas atividades de maneira visual (NETO, 2017).

Presente nas metodologias ágeis: Lean.

Estudo na literatura que menciona a prática abordada: Modelo de gerenciamento ágil de projetos utilizando a metodologia kanban: aplicação em uma empresa de software, 2020.

Modelo para Utilização da Prática: Para a utilização dessa prática a equipe deverá adotar um quadro com quatro colunas: A fazer, fazendo, testando e feito. Em seguida deverão inserir as atividades da equipe em cartões de acordo com cada coluna, posteriormente a medida que as tarefas forem realizadas os cartões serão movidos entre as colunas.

3.22. Lead time (Tempo de Espera)

Essa prática é conhecida como o tempo de espera que vai desde a identificação de um requisito até sua implementação (HOST; REGNELL; WOHLIN, 2000). Este termo faz referência aos métodos de fabricação conhecidos como Lean ou Toyota Production System, onde este tempo é definido como o tempo gasto entre um cliente fazer um pedido e receber o produto solicitado (AGILE ALLIANCE, 2022).

Principais características e benefícios: Com a utilização dessa prática será possível medir o tempo de efetividade das entregas dentro do projeto, além disso, a equipe poderá verificar quais etapas estão demandando mais tempo para ser entregues (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Lean.

Estudo na literatura que menciona a prática abordada: Metodologias ágeis: análise e comparação do Scrum, Kanban e Lean aplicados ao desenvolvimento de software, 2021.

Modelo para Utilização da Prática: Para a utilização dessa prática deverá ser medido o tempo gasto entre a identificação dos requisitos até propriamente a sua implementação. Um ponto de atenção, e que a métrica utilizada assim como o período deverá ser acordado previamente com todos os membros da equipe. Após a medições os dados poderão ser exibidos em gráficos para uma melhor compreensão de todos.

3.23. Desenvolvimento iterativo

Essa prática está aliada ao desenvolvimento incremental, onde as atividades poderão ser revisitadas em diversos momentos do projeto, e é organizado em iterações com objetivos determinados (MARTIN, 1999).

Principais características e benefícios: Uma das principais ferramentas dessa prática está na possibilidade de prototipagem, pois a mesma tem uma estratégia iterativa e pode ser um precursor para o desenvolvimento de ideias iterativas de desenvolvimento de software (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Scrum e Fundamentals.

Estudo na literatura que menciona a prática abordada: Uma abordagem para modelagem de software utilizando a OPM para desenvolvimento iterativo, incremental e ágil, 2017.

Modelo para Utilização da Prática: Para a utilização dessa prática a equipe deverá utilizar a metodologia de desenvolvimento iterativo, onde cada incremento do projeto deverá ser desenvolvido, testado, integrado e validado com cliente.

3.24. Timebox

Essa prática determina que a equipe deverá fixar uma quantidade de tempo a ser utilizada para o desenvolvimento das etapas do software, ou seja, o timebox é um intervalo de tempo pré-definido no projeto, tendo como entrada a lista de funções e como saída o software que será avaliado para decidir se está pronto para ser colocado em produção (ABBAS; GRAVELL; WILLS, 2008).

Principais características e benefícios: Um dos benefícios dessa prática é a possibilidade de interromper o trabalho quando o limite de tempo for atingido e avaliar o que foi realizado e quais correções nos processos deverão ser feitas. (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Scrum.

Estudo na literatura que menciona a prática abordada: Uma ferramenta web para suporte à definição de processos ágeis, 2013.

Modelo para Utilização da Prática: Para a realização dessa prática deverá ser definido um período de tempo previamente acordado junto aos membros da equipe, afim de trabalharem de forma constante para a conclusão do objetivo definido no início do processo (AGILE ALLIANCE, 2022).

3.25. Três perguntas

Essa prática orienta que durante a reunião diária sejam realizadas três perguntas “O que você concluiu desde a última reunião?”, “O que você planeja finalizar até a próxima reunião?”, “Existe algum impedimento?” (SUTHERLAND; VIKTOROV, 2007).

Principais características e benefícios: Nessa prática uma das principais características é entender o andamento das tarefas da equipe por meio das respostas de cada indivíduo e posteriormente adotar medidas para melhorar o desempenho de cada um.

Presente nas metodologias ágeis: Scrum.

Estudo na literatura que menciona a prática abordada: Uso de metodologias de desenvolvimento de software e de engenharia de requisitos em empresas de tecnologia: um estudo a partir de um survey, 2019.

Modelo para Utilização da Prática: Para a realização dessa prática é necessário que cada membro da equipe responda as seguintes perguntas: O que você fez desde a última reunião; O que você planeja fazer até a próxima reunião; existe algum impedimento. Mediante as respostas dos membros da equipe os gestores ou líderes poderão adotar medidas para melhorarem a performance de cada um.

3.26. Gráfico de Burndown

Essa prática recomenda que seja exibido um gráfico onde o eixo vertical representa a quantidade de trabalho restante e no eixo horizontal o tempo decorrido desde o início do projeto exibindo o futuro e o passado. O gráfico é considerado um “radar de informações”, desde que os mesmos sejam frequentemente atualizados (AGILE ALLIANCE, 2022).

Principais características e benefícios: O principal benefício dessa prática está na possibilidade de visualização do status atualizado do projeto, ou seja, o gráfico incentiva a equipe a enfrentar quaisquer dificuldades mais cedo e de forma mais decisiva. Um ponto importante é que o mesmo deverá ser grande o suficiente e situado em algum lugar que possa proporcionar discussões entre os membros da equipe. Além disso, o mesmo não pode ficar apenas em uma folha de papel A4 em um corredor distante ou no fundo de uma gaveta. Outro ponto que merece destaque é a simplicidade na criação dos gráficos e na sua eficácia (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Scrum.

Estudo na literatura que menciona a prática abordada: Estudo da aplicação de metodologia ágil para acelerar o processo da maturidade em processos de TI na pequena empresa, 2019.

Modelo para Utilização da Prática: Para a realização dessa prática a equipe deverá criar um gráfico com tempo e entregas realizadas e posteriormente disponibilizar em um local bem visível para todos os membros da equipe. O mesmo deverá ser atualizado frequentemente para uma análise mais assertiva dos dados.

3.27. Quadro de Tarefas

Prática que recomenda a utilização de um quadro branco ou até mesmo em uma parte da parede, o mesmo deverá ser dividido em três colunas: “A Fazer”, “Em Andamento” e “Concluído, posteriormente notas adesivas deverão ser colocadas nas colunas refletindo o status atual das tarefas (AGILE ALLIANCE, 2022).

Principais características e benefícios: Os benefícios mais relevantes estão na simplicidade e na flexibilidade do quadro de tarefas e seus materiais elementares como notas adesivas, pontos adesivos. Isso permite que a equipe possa representar qualquer informação relevante no quadro. Além disso, cores poderão ser usadas para distinguir recursos de correções de bugs ou andamentos da atividade (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Scrum.

Estudo na literatura que menciona a prática abordada: Não foram encontrados na literatura documentos que relatam ou mencionam a utilização desta prática.

Modelo para Utilização da Prática: Para a realização dessa prática deverá ser disponibilizado um quadro com três colunas “A Fazer”, “Em Andamento” e “Concluído”. Posteriormente as tarefas da equipe deverão ser inseridas por meio de notas adesivas de acordo com o status disponíveis nas colunas.

3.28. Definição de Pronto

Essa prática pode ser entendida como uma atividade que foi concluída no projeto (SOUZA, 2021).

Principais características e benefícios: Com a definição de pronto a equipe consegue realizar uma estimativa dos pontos concluídos no projeto, evitando mal-entendidos na comunicação e melhorando a performance das entregas no projeto (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Scrum e Lean.

Estudo na literatura que menciona a prática abordada: Metodologia ágil para gestão e planejamento de projetos: implantação do Scrum e do framework Jira na qualidade em uma empresa do setor aeronáutico, 2018.

Modelo para Utilização da Prática: Para a realização dessa prática a medida que o incremento ou fase do projeto for entregue o mesmo deverá literalmente “PRONTO” ou seja pronto para a sua utilização.

3.29. Estimativa de Pontos

Prática que utiliza uma das unidades, mas difundidas nas metodologias ágeis, que são os “pontos da história” (AGILE ALLIANCE, 2022).

Principais características e benefícios: Essa prática tem como característica principal o uso das unidades como pontos de história, ela enfatiza a dificuldade relativa sobre a o tempo gasto, a velocidade absoluta. Portanto alivia algumas das tensões que geralmente surgem entre os desenvolvedores e gerentes em torno da estimativa do projeto: por exemplo, solicitar aos desenvolvedores uma estimativa, e logo, responsabilizando-os como se fosse um compromisso firme (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Scrum.

Estudo na literatura que menciona a prática abordada: Fatores críticos para aplicação do Scrum em projetos de desenvolvimento de software, 2022.

Modelo para Utilização da Prática: Para a realização dessa prática deverão ser mapeadas os pontos da história e em seguida mensurados afim de analisar como está o andamento das atividades.

3.30. Estimativa Relativa

Prática ágil que realiza uma estimativa não absoluta de tempo das histórias de usuários (tarefas) em comparação com outras histórias de usuários (MIRANDA, 2001).

Principais características e benefícios: Uma das características importantes dessa prática está no fato de algumas equipes usarem escalas não numéricas como forma de “forçar” a estimativa relativa e os nomes das técnicas correspondentes refletem a escala, onde são comuns a utilização de escalas mais exóticas, por exemplo, frutas, bem provável que seja mais pelo valor de novidade do que por quaisquer ganhos reais em clareza (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Scrum.

Estudo na literatura que menciona a prática abordada: Proposta de reestruturação do processo de estimativa de tarefas em projetos de software numa empresa que utiliza o método ágil Scrum, 2014.

Modelo para Utilização da Prática: Para a realização dessa prática deverão ser realizadas as estimativas não absolutas das histórias de usuários.

3.31. Poker de Planejamento:

Essa prática utiliza uma abordagem para a realização de uma estimativa, onde o cliente ou o product Owner descreve uma funcionalidade para os estimadores, esses por sua vez seguram uma carta de baralho, geralmente seguindo a sequência de Fibonacci (0, 1, 2, 3, 5, 8, 13, 20, 40 e 100). Esses valores geralmente representam as unidades de estimativas que fazem sentido para o time (SCHWABER e SUTHERLAND, 2009).

Principais características e benefícios: A característica que chama mais atenção nessa prática está no formato em que a reunião é realizada. Pois trata-se de uma oportunidade para alavancar a troca de conhecimentos entre os membros da equipe, tendo em vista que em um formato de reunião menos estruturado, os membros mais participativos às vezes excluem os mais retraídos. Além disso, é uma ótima maneira de reunir insights sobre a história do usuário que está sendo discutida e expor os desafios da implementação (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Scrum.

Estudo na literatura que menciona a prática abordada: Ferramenta Open-Source para apoio ao uso do Scrum por equipes distribuídas, 2009.

Modelo para Utilização da Prática: Para a utilização dessa prática o product owner deverá resumidamente falar sobre o valor e a intenção de uma história. Em seguida, cada membro da equipe de desenvolvimento seleciona uma estimativa e prepara a carta correspondente, virada para baixo. Quando todos os membros tiverem já escolhidos as suas respectivas cartas, as mesmas são viradas para cima e as estimativas são lidas em voz alta. Os membros da equipe que escolheram a estimativa mais alta e mais baixa justificam seu raciocínio. Depois de uma breve discussão entre os integrantes, a equipe pode buscar a convergência para uma estimativa de consenso jogando uma ou mais rodadas adicionais (AGILE ALLIANCE, 2022).

3.32. Backlog

A prática consiste em uma lista ordenada tudo que precisa ser desenvolvido, modificado ou corrigido no projeto (SCHWABER; SUTHERLAND, 2009). Além disso, pode ser considerado como uma lista de novos recursos, correções de bugs e alterações de infraestrutura ou em outras atividades que seja necessário realizar uma entregar para alcançar um resultado específico (AGILE ALLIANCE, 2022).

Principais características e benefícios: Essa prática é caracterizada por ser a única fonte autorizada para as funcionalidades em que a equipe poderá trabalhar. Basicamente isso significa que nada será feito se não estiver no backlog. Porém, o simples fato da presença de um item em um backlog do produto não garante que ele será entregue (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Scrum.

Estudo na literatura que menciona a prática abordada: Adoção de SCRUM em uma fábrica de desenvolvimento distribuído de software, 2007.

Modelo para Utilização da Prática: Para a realização dessa prática a equipe deverá definir os itens que farão parte do backlog, podendo iniciar a lista com as histórias de usuários. O formato e estilo a serem utilizados deve ser previamente alinhado com todos os membros da equipe.

3.33. Desenvolvimento incremental

Prática ágil onde o projeto é dividido em pequenas etapas (incrementos) e são adicionadas novas funcionalidades visíveis aos usuários em cada versão do projeto (MARTIN, 1999).

Principais características e benefícios: Essa prática tem como característica principal a entrega de um produto operacional a cada incremento, ou seja, uma funcionalidade do produto sem erros e pronto para ser utilizado pelo usuário. Portanto mesmo que os incrementos iniciais sejam partes do produto, essas partes são operacionais e funcionam sem as outras (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Product Manager e Fundamentals.

Estudo na literatura que menciona a prática abordada: Metodologias ágeis para o desenvolvimento de softwares, 2018.

Modelo para Utilização da Prática: Para a realização dessa prática deverão ser acrescentadas novas funcionalidades ao sistema à medida que o software é desenvolvido ou durante a evolução do mesmo. Na primeira opção, as funcionalidades são implementadas completamente e entregues uma por vez, no segundo caso elas são criadas de forma simplificada para rapidamente entrarem em produção e, caso seja necessário outras funcionalidades serão acrescentadas na próxima (Iteração) versão (BASSI FILHO, 2008).

3.34. INVEST

Essa prática utiliza o acrônimo INVEST que em sua tradução literal significa (Independente, Negociável, Avaliável, Estimável, Apropriadamente Dimensionada e Testável) (LONGO E SILVA, 2014). A utilização do INVEST ajuda a refinar as funcionalidades do software e validar decisões no processo. Sendo utilizada principalmente para validação das histórias de usuários (AGILE ALLIANCE, 2022).

Principais características e benefícios: Entre os principais benefícios do uso do INVEST está em contribuir para um melhor entendimento dos requisitos do sistema, realizar estimativas mais acertada, estabelecer critérios de aceite e determinar o valor agregado do requisito do negócio priorizando o que realmente de ser entregue ao cliente (WAKE, 2003).

Presente nas metodologias ágeis: Product Manager.

Estudo na literatura que menciona a prática abordada: A utilização de histórias de usuários no levantamento de requisitos ágeis, 2014.

Modelo para Utilização da Prática: Para a realização dessa prática deverão ser obedecidos os critérios definidos na INVEST, ou seja, a mesma deverá ser independente, negociável, avaliável, estimável, apropriadamente dimensionada e testável. Podendo utilizar essa prática nas histórias do usuário, para refinar as funcionalidades do software e auxiliar na validação de decisões no processo de desenvolvimento (AGILE ALLIANCE, 2022).

3.35. Os Três Cs

Prática ágil que estabelece três critérios básicos Cartão, Conversa e Confirmação. O cartão serve como base para o início de uma conversa onde todos os detalhes são discutidos e o resultado desse discurso, ou seja, os critérios de aceitação dessa história são elencados e anotados em um documento de confirmação (SHARP; ROBINSON; PETRE, 2009).

Principais características e benefícios: Essa prática colabora com um conjunto de critérios amplamente aceito, ou checklist, que auxilia a avaliar a qualidade de uma história de usuário. Caso a história não atenda a um desses critérios, a equipe de desenvolvimento poderá reformulá-la ou até mesmo considerar uma reescrita, o que

geralmente significa rasgar fisicamente o cartão de história antigo e escrever um novo (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Product Manager.

Estudo na literatura que menciona a prática abordada: Concepção de aplicações de software de Business Analytics com metodologias ágeis, 2017.

Modelo para Utilização da Prática: Para a realização dessa prática basicamente a equipe deverá contar com 03 elementos: Cartão (habitualmente utilizado um post-it ou cartão de anotação), Conversação (Conversa entre os membros do projeto) e Confirmação (Geralmente utilizado um documento de aceite).

3.36. Divisão de Histórias

Prática ágil que recomenda a divisão das histórias, ou seja, ela deverá ser dividida em uma série de histórias menores e deverão ser implementadas em até 03 dias (PATTON, 2008).

Principais características e benefícios: O principal benefício dessa prática está justamente no propósito de apoiar e transmitir a necessidade do usuário de utilizar uma linguagem simples e clara, facilitando o entendimento entre as partes. Dessa forma o nível de entendimento e assertividade no desenvolvimento do produto é bem maior (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Product Manager

Estudo na literatura que menciona a prática abordada: Modelo de desenvolvimento de objetos de aprendizagem baseado em metodologias ágeis, 2009.

Modelo para Utilização da Prática: Para a realização dessa prática a história de usuário deverá ser subdividida em várias histórias menores, o prazo de implementação é de no máximo 03 dias. Caso não seja possível a implementação no prazo estimado a história deverá ser dívida novamente afim de cumprir os prazos.

3.37. Mapeamento de história

Prática ágil que orienta que as histórias dos usuários sejam divididas em duas vertentes, uma na horizontal com sequencia comportamental do sistema e outra na vertical com os incrementos referentes as evoluções das funções do sistema (PATTON, 2005).

Principais características e benefícios: Tem como característica principal a criação de um mapa com todas histórias organizadas em grupos funcionais. Tal prática ajuda a manter visualmente um panorama geral e ao mesmo tempo busca fornecer os todos os detalhes de determinada aplicação (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Product Manager.

Estudo na literatura que menciona a prática abordada: Modelo de desenvolvimento de objetos de aprendizagem baseado em metodologias ágeis e Scaffoldings, 2010.

Modelo para Utilização da Prática: Para a realização dessa prática deverá ser realizado um mapeamento das histórias afim de identificar e ordenar cada uma de acordo com o eixo vertical ou horizontal, descritos anteriormente.

3.38. Personas

Prática que utiliza um modelo de um usuário, descrito detalhadamente em termos de necessidades, objetivos e tarefas (AOYAMA, 2005). Quando em um projeto é considerado importante a experiência de um usuário a equipe cria biografias detalhadas e sintéticas de usuários fictícios do produto que será desenvolvido, esses usuários fictícios são chamados de “personas” (AGILE ALLIANCE, 2022).

Principais características e benefícios: A característica principal dessa prática está na possibilidade da criação de personagens hipotéticos bem específicos, ligados principalmente aos detalhes sobre os hábitos, trabalho e consumo de um ou mais dos

seus clientes alvo, a fim de gerar maiores detalhes e insights durante o desenvolvimento do produto (KOTLER et al., 2012)

Presente nas metodologias ágeis: Product Manager.

Estudo na literatura que menciona a prática abordada: Proposta de um método estruturado para a construção de uma solução de espaços inteligentes por meio de práticas ágeis, 2020.

Modelo para Utilização da Prática: Para a elaboração dessa prática deverá ser criado personagens fictícios para representar os possíveis usuários do sistema, durante a criação do personagem a equipe deverá ter atenção nos detalhes sobre os hábitos, trabalho e consumo do seu cliente alvo.

3.39. Preparação de Pendências

Prática também conhecida como refinamento de pendências. A mesma consiste em uma revisão dos itens da pendência para garantir que a mesma contenha os itens apropriados, que sejam priorizados (AGILE ALLIANCE, 2022).

Principais características e benefícios: O principal benefício dessa prática está em garantir que o backlog permaneça preenchido com itens relevantes, detalhados e estimados de acordo com a sua prioridade e entendimento do projeto. Por exemplo, nem todas as histórias de usuários tem a necessidade de ser divididas em um nível refinado no início do projeto. No entanto é importante que a qualquer momento um número “suficiente” de histórias esteja pronto para agendamento nas próximas iterações (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Product Manager.

Estudo na literatura que menciona a prática abordada: Não foram encontrados na literatura documentos que relatam ou mencionam a utilização desta prática.

Modelo para Utilização da Prática: Para a realização dessa prática algumas das atividades deverão ser realizadas, por exemplo, remover histórias de usuários que não são mais relevantes para o projeto, criar novas histórias de usuário de acordo com às necessidades descobertas durante a evolução do projeto e reavaliar a prioridade relativa das histórias (AGILE ALLIANCE, 2022).

3.40. Controle de Versão

Prática que utiliza um controle de versão, para criar, armazenar e reconstruir versões do sistema de software (BALL; KIM; SIY, 1997).

Principais características e benefícios: Um dos principais benefícios dessa prática está em refletir detalhadamente sobre suas políticas de infraestrutura de controle de versão e garantir que as práticas de engenharia funcionem corretamente (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Devops e Fundamentals.

Estudo na literatura que menciona a prática abordada: Analisando a agilidade em processos ágeis, 2011.

Modelo para Utilização da Prática: Para a realização dessa prática o time de desenvolvimento deverá contar com mecanismo para o controle de versão, uma das ferramentas mais utilizadas é a plataforma GitHub, a mesma realiza a hospedagem de código, controle de versão e possibilita que mais de um desenvolvedor trabalhe junto em um mesmo projeto (GITHUB, 2022).

3.41. Compilação Automatizada

Prática ágil onde as etapas repetíveis poderão ser automatizadas, ou seja, não necessitam diretamente de intervenção humana e podendo ser executadas a qualquer instante, sem

nenhuma informação adicional, além do que já está armazenado no repositório de controle de código-fonte (PONTES E ARTHAUD, 2018).

Principais características e benefícios: Entre as principais características estão a possibilidade da criação de automação entre as tarefas, por exemplo, um determinado software poderia extrair e organizar os resultados de uma pesquisa, apresentando posteriormente ao gestor. Dessa forma o processo de decisão se torna mais ágil e eficiente (PONTES E ARTHAUD, 2018).

Estudo na literatura que menciona a prática abordada: Estudo de caso: processos de desenvolvimento de software embarcado utilizando ferramentas de automatização, 2021.

Presente nas metodologias ágeis: Devops.

Modelo para Utilização da Prática: Para a realização dessa prática a equipe deverá planejar e implantar a automação das tarefas que podem ser automatizadas, deixando o projeto mais rápido e eficiente.

3.42. Implantação contínua

Prática que é considerada uma extensão da integração contínua, ou seja, a mesma consiste em automatizar o processo de implantação do software (HUMBLE; READ; NORTH, 2006). Nessa prática a equipe conta com uma infraestrutura que instrumenta e automatiza as diversas etapas que levam à implantação, afim de que, depois de cada integração que atenda com sucesso os critérios de lançamento, o sistema atual seja atualizado com novo código (AGILE ALLIANCE, 2022).

Principais características e benefícios: Os principais benefícios dessa prática estão na redução do lead time, além do retorno antecipado do investimento para cada recurso depois de desenvolvido, e a redução na necessidade de grandes investimentos de capital. Outro benéficos está no feedback prévio dos usuários sobre cada nova funcionalidade à medida que é lançada para produção (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Devops

Estudo na literatura que menciona a prática abordada: Mitigação de dificuldades na implantação de métodos ágeis de desenvolvimento de software no SERPRO, 2015.

Modelo para Utilização da Prática: Para realização dessa prática a equipe deverá realizar uma implantação contínua de novas funcionalidades do sistema.

3.43. Cartões CRC

Nessa prática são utilizados cartões contendo as classes, os colaboradores e as suas responsabilidades. Essa prática tem a intenção de esboçar rapidamente diversas ideias diferentes para algum recurso do projeto (AGILE ALLIANCE, 2022).

Principais características e benefícios: O principal benefício dessa prática está na possibilidade da reutilização de funcionalidades e classes disponíveis no sistema.

Presente nas metodologias ágeis: Designer

Estudo na literatura que menciona a prática abordada: Estudo das subáreas da engenharia de requisitos em metodologias ágeis, 2013.

Modelo para Utilização da Prática: Para a realização dessa prática os membros da equipe anotam os nomes das classes mais importantes envolvidas na funcionalidade do sistema. Em seguida os cartões são preenchidos com as listas de responsabilidades de cada classe e posteriormente os nomes dos colaboradores. Por fim, o próximo passo é validar – ou invalidar, cada ideia de design, ou seja, cada desenvolvedor assumi o papel de uma ou mais classes (AGILE ALLIANCE, 2022).

3.44. Sessão de Design Rápido

Essa prática projeta e prioriza a solução mais simples possível e que tenha a intenção de solucionar apenas os problemas atuais, outro ponto importante, e que a complexidade e o código extras devem ser evitados (BECK, 2004).

Principais características e benefícios: A principal característica dessa prática está em favorecer um “design simples”, aos desenvolvedores que geralmente lidam com as decisões locais de design a todo momento, porém estão alertas para escolhas de design que podem ser de longo alcance. Quando tal escolha surge, dois ou mais desenvolvedores se reúnem para uma breve sessão de design, ou seja, a prática de design simples, recomenda que o time opte por um código mais simples, que seja suficiente para atender às necessidades atuais (BECK, 2000).

Presente nas metodologias ágeis: Designer.

Estudo na literatura que menciona a prática abordada: Análise da aderência das práticas recomendadas pelos métodos ágeis: XP, Scrum e TDD com a utilização de ferramentas CASE, 2012.

Modelo para Utilização da Prática: Para a utilização dessa prática deverão ser seguidas as seguintes etapas: o código deverá ter um design simples e funcional, ser de fácil compreensão por todos os membros da equipe de desenvolvimento, o design deve atingir o objetivo para que foi criado, as repetições de códigos idênticos deverão ser evitadas e não deve ser criada funcionalidades extras que não serão utilizadas naquele momento (POPPENDIECK & POPPENDIECK, 2003).

3.45. Regras de Simplicidade

Essa prática consiste na realização de testes referente a regras de simplicidade do código, onde o sistema frequentemente verifica se existe duplicidade de códigos e se o mesmo está utilizando o mínimo de classes possíveis (BECK, 2000).

Principais características e benefícios: O principal benefício dessa prática estão na simplicidade e na comunicação pois existe uma excelente relação de apoio mútuo. Portanto, quanto maior for o grau de comunicação, mais claramente será possível ver o que precisa ser feito e maior será a certeza sobre o que realmente não precisa ser feito. Dessa forma, quanto mais simples for o seu sistema, menor será a necessidade de comunicar sobre ele (BECK, 2000).

Presente nas metodologias ágeis: Designer.

Estudo na literatura que menciona a prática abordada: Métodos ágeis e gestão de software livre, 2019.

Modelo para Utilização da Prática: Para a realização dessa prática deverá ser criada rotinas de verificação do código, referentes a simplicidade, a duplicidade do código e a quantidade no número de classes utilizadas.

3.46. Linguagem Onipresente (ubíqua)

Prática ágil usada para realizar a comunicação entre os membros da equipe, a mesma deve conter detalhes sobre o domínio do tipo de negócio combinados com o os nomes padrões de design de sistemas utilizados (LANDRE; WESENBERG; OLMHEIM, 2007).

Principais características e benefícios: Uma das características mais importantes está no constante conflito introduzido pela utilização de dois vocabulários distintos, de um lado a linguagem do tipo de negócio e do outro lado a linguagem dos desenvolvedores. Até determinado ponto, essa duplicidade é inevitável, pois os desenvolvedores devem adequar seus trabalhos em termos de algoritmos de computação, que habitualmente não têm equivalência direta no vocabulário de negócios.

Presente nas metodologias ágeis: Designer.

Estudo na literatura que menciona a prática abordada: uma abordagem ágil para transformar modelos cognitivos em modelos comportamentais e de domínio, 2014.

Modelo para Utilização da Prática: Para a utilização dessa prática a equipe deverá definir qual vocabulário será utilizado em um determinado projeto de acordo com tipo de negócio e design utilizado no projeto.

3.47. Testando usabilidade

Essa prática utiliza uma técnica exploratória e empírica criada a bastante tempo para responder a perguntas como: Um usuário final conseguiria utilizar o software desenvolvido em condições realistas? (AGILE ALLIANCE, 2022).

Principais características e benefícios: A principal característica dessa prática está em analisar um usuário final representativo utilizando o produto, em um determinado objetivo a ser alcançado, porém sem muitas instruções específicas para usar o produto. Justamente para testar a usabilidade e facilidade em utilizar a funcionalidade desenvolvida (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Testing.

Estudo na literatura que menciona a prática abordada: Uma estratégia de apoio à institucionalização da usabilidade em ambientes de desenvolvimento ágil, 2008.

Modelo para Utilização da Prática: Para a utilização dessa prática a equipe poderá solicitar a um usuário que não tenha participado do desenvolvimento do sistema, que realize análise da usabilidade e facilidade das funcionalidades desenvolvidas no sistema.

3.48. Teste exploratório

Essa prática utiliza diferentes formas de realizar determinado teste, a mesma da possibilidade para que sejam criados novos e melhores testes a partir do que já foi criado, ela não se limita a apenas com testes de scripts (BACH, 2003).

Principais características e benefícios: A principal característica dessa prática está relacionada com a realização de várias atividades relacionadas aos testes, por exemplo, o projeto de teste, a execução de teste e interpretação de resultados dos testes, sendo realizados de forma intercalada, no decorrer do projeto, em vez de em uma sequência fixa (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Testing.

Estudo na literatura que menciona a prática abordada: Framework para testes ágeis de software: uma proposta exploratória, 2018.

Modelo para Utilização da Prática: Para a utilização dessa prática deverão ser elaborados diversos teste das funcionalidades do software, a equipe pode definir um testador para realização dessas atividades ao longo do projeto, o mesmo deverá ter uma participação ativa no time, entendendo as integrações e a automação das atividades afim de testar da melhor forma possível.

3.49. Teste de unidade

Essa prática proporcionam uma rede de segurança dos testes de regressão e validação, para que seja possível refatorar e integrar de forma eficaz. Além disso, a criação de testes de unidades antes do código auxilia ainda mais a consolidar os requisitos aumentando o foco do desenvolvedor (BECK, 2000).

Principais características e benefícios: O principal benefício dessa prática está na possibilidade da realização de testes dos componentes do software, por exemplo, como métodos ou classes de objeto, pois estes são o tipo mais simples de componente. Além disso, esses testes priorizam a lógica interna de processamento e as estruturas de dados nos limites de um componente (JOVANOVIĆ, 2008; WAZLAWICK, 2013).

Presente nas metodologias ágeis: Testing.

Estudo na literatura que menciona a prática abordada: A importância dos testes automatizados, 2008.

Modelo para Utilização da Prática: Para a realização dessa prática a equipe deverá examinar o comportamento de uma unidade distinta de trabalho, realizando o teste unitário, o mesmo deverá assegurar que as unidades menores: módulo, método ou classe funcionem de acordo com o que foi especificado (MASSOL et al., 2010).

3.50. Objetos simulados

Essa prática consiste na realização de uma técnica usada habitualmente no contexto de desenvolvimento de testes unitários automatizados. Ou seja, ela instancia uma versão específica de teste de um componente de software, geralmente uma classe, que diferente dos comportamentos normais, fornece resultados pré-computados (AGILE ALLIANCE, 2022).

Principais características e benefícios: A principal característica dessa prática está na possibilidade de separar os componentes de software para fins de teste, por exemplo, o código que será inserido, não necessariamente deverá observar os detalhes de implementação de banco de dados, sua correta função depende muito mais da organização lógica dos dados e não de aspectos concretos, como o armazenamento dos dados em disco ou sobre estabelecer os protocolos de conexões em rede. Portanto, os objetos simulados realizam testes que apenas usam o comportamento dos componentes de software e não dependem do comportamento dos outros componentes não relacionados (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Testing.

Estudo na literatura que menciona a prática abordada: WS-TDD - Uma abordagem ágil para o desenvolvimento de serviços web, 2016.

Modelo para Utilização da Prática: Para a utilização dessa prática deverão ser definidos os testes com os objetos simulados do software, de acordo com o exemplo acima descrito, a equipe deverá selecionar quais componentes serão testados.

3.51. Teste de aceitação

Essa prática consiste em uma descrição formal do comportamento de um produto de software. Na maioria das vezes é utilizado um exemplo ou cenário de uso. Geralmente, o objetivo é que seja possível automatizar a execução dos testes por uma ferramenta de software (AGILE ALLIANCE, 2022).

Principais características e benefícios: Uma das principais características dessa prática é desenvolver um processo para a definição dos critérios de aceitação. Além de criar um processo para a elaboração de um plano de teste de aceitação e desenvolver um processo para o registo dos testes e posteriormente apresentar os resultados dos testes de aceitação (MAIDASANI, 2007).

Presente nas metodologias ágeis: Testing.

Estudo na literatura que menciona a prática abordada: Métricas para medir a uniformidade de dados em teste de aceitação, 2020.

Modelo para Utilização da Prática: Para a realização dessa prática a equipe deverá elaborar os testes de aceitação das funcionalidades que estão sendo desenvolvidas, ou seja, serão verificados se todos os itens estão de acordo como foram definidos. Após a realização dos testes, o mesmo deverá ser registrado e o documento de aceite assinado.

3.52. Desenvolvimento Orientado a Testes de Aceitação (ATDD)

Essa prática enfatiza que definindo os testes de aceitação e discutindo sobre eles detalhadamente haverá um melhor entendimento dos requisitos. Isso acontece devido os testes de aceitação exigirem que se chegue a um acordo sólido sobre o comportamento exato que o software deverá expor quando estiver concluído (HENDRICKSON, 2008).

Principais características e benefícios: A principal característica dessa prática está em adotar uma metodologia de desenvolvimento de software que promove a participação entre desenvolvedores, testadores e clientes, na qual a automação dos testes desempenha um papel de destaque (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Testing.

Estudo na literatura que menciona a prática abordada: Estudo sobre metodologias ágeis de desenvolvimento aplicando a metodologia extreme programming em uma aplicação web, 2014.

Modelo para Utilização da Prática: Para a realização dessa prática todos os envolvidos no projeto deverão se unir, colaborando entre si, para escreverem os testes de aceitação antes da implementação das funcionalidades do software (GARTNER, 2012).

3.53. Desenvolvimento Orientado ao Comportamento (BDD)

Essa prática consiste na automatização de testes de comportamentos esperados do software, uma vez que estes comportamentos deverão descrever os requisitos do sistema (TAVARES et al., 2010).

Principais características e benefícios: Essa prática tem como principal característica aumentar o entendimento da equipe em relação ao comportamento do software que será desenvolvido. Além de possibilitar as práticas como o Test-Driven Development (TDD) (ASTELS, 2003).

Presente nas metodologias ágeis: Testing.

Estudo na literatura que menciona a prática abordada: ProDOC - Uma proposta de processo de desenvolvimento orientado a comportamento, 2018.

Modelo para Utilização da Prática: Para a utilização dessa prática a equipe deve ser capaz de fornecer uma parte da documentação funcional na forma de histórias de usuário, se possível com cenários ou exemplos executáveis. Além disso, ao invés de se referir a teste, deverão preferir utilizar o termo cenário e especificação. Portanto, deverão reunir em um único local a especificação de um resultado importante para o usuário, geralmente usando a matriz de funções (AGILE ALLIANCE, 2022).

3.54. Dado – Quando – Então

Essa prática consiste em um modelo para escrita de testes de aceitação, buscando evitar o excesso de textos dado um contexto, quando um evento ocorre, então um resultado é esperado (TON, 2007).

Principais características e benefícios: A principal característica dessa ferramenta está na utilização da técnica Dado-quando-então, sendo está um modelo destinado a orientar a realização de testes de aceitação para uma história de usuário (AGILE ALLIANCE, 2022).

Presente nas metodologias ágeis: Testing.

Estudo na literatura que menciona a prática abordada: Proposta de uso da técnica BDD para otimizar a escrita e automação de testes no framework Scrum, 2019.

Modelo para Utilização da Prática: Para a utilização dessa prática serão seguidos basicamente três elementos: Algum contexto (Dado); alguma ação é realizada (Quando); um conjunto particular de consequências observáveis deve-se obter (Então) (AGILE ALLIANCE, 2022).

3.55. Modelo de história de usuário

Essa prática consiste na utilização de formato comum utilizado para a escrita de história de usuário, que auxilia na inclusão de informações importantes sobre essa história. E a prática mais recomendada para auxiliar as equipes e proprietários de produtos que querem começar a trabalhar com histórias de usuários (AGILE ALLIANCE, 2022).

Principais características e benefícios: As principais características dessa prática estão as descrições das histórias de usuários, que devem ser escritas de maneira curta e abstrata do que deve ser implementado e que primeiramente devem servir como âncoras para maiores discussões com os clientes (MAURER; MARTEL, 2002).

Presente nas metodologias ágeis: Testing.

Estudo na literatura que menciona a prática abordada: Experiências de ensino a distância do gerenciamento ágil de projetos com Scrum e apoio de uma ferramenta para gerência de histórias de usuário, 2016.

Modelo para Utilização da Prática: Para a realização dessa prática a equipe deverá escrever histórias de usuário e maneira simples e objetiva. O principal objetivo e especificar as funcionalidades do sistema da melhor maneira possível, sem entrar nos detalhes técnicos ou de implementação, qual o propósito da análise, quem são os envolvidos e o contexto na qual a pesquisa está inserida.

4. CONCLUSÃO

Durante a elaboração desse estudo percebeu-se que, mesmo existindo um grande aumento na utilização das práticas ágeis, existem pontos fracos com relação a elas. Um dos principais exemplos que se pode citar está na falta de documentação consistente sobre o tema abordado, ou ainda, a falta de referências bibliográficas relatando detalhes de determinadas práticas ágeis, ou seja, existem estudos considerados superficiais quando comparados com a bibliografias de modelos tradicionais de desenvolvimento de software. (PONTES E ARTHAUD, 2018).

No entanto, a capacidade natural das práticas ágeis em se adaptar a cada projeto em que ela será utilizada, pode fazer com que as percepções dessas falhas não sejam sentidas pela equipe, tendo em vista a utilização de mais de uma prática em cada projeto, ou até mesmo a utilização parcial de determinada prática afim de otimizar um determinado processo de software. Entretanto, assim como nas metodologias tradicionais, não existe fórmula mágica para a realização dos processos de desenvolvimento de software. O sucesso na aplicação das práticas depende do conhecimento da equipe aliadas experiência e a melhoria contínua do projeto (PONTES E ARTHAUD, 2018).

Além disso, através do presente estudo, observou-se que as práticas ágeis catalogadas em sua maioria foram desenvolvidas afim de otimizar o processo de desenvolvimento de software, para satisfazer clientes que queriam uma versão funcional do produto o mais rápido possível, e isto só foi possível com a criação de novas técnicas ágeis, onde inúmeros processos foram melhorados e houve um aumento na satisfação do cliente.

Ademais, constatou-se que não existe um consenso sobre quais práticas são mais recomendadas em um determinado cenário, ou seja, depende da análise criteriosa da equipe em decidir qual prática adotar no processo e quais serão os benefícios da mesma. Pode-se ainda destacar que estudos mostram que algumas práticas são poucos relevantes para o andamento do processo de software, por outro lado, o mesmo estudo afirma que, as características de trabalho em equipe, adaptabilidade a mudanças e a implementação de constantes feedback, foram consideradas de alta relevância em todas as práticas ágeis. (DE MELLO, DA SILVA, TRAVASSOS, 2014).

Portanto, a partir da catalogação das práticas ágeis, foi possível observar o quanto é difícil responder o que realmente caracteriza agilidade no processo de desenvolvimento de

software, principalmente quando se trata de uma heterogeneidade na comunidade de engenharia de software (DE MELLO, DA SILVA, TRAVASSOS, 2014). Onde muitos especialistas no assunto defendem suas práticas sem a observância no processo como um todo. Dessa forma, é importante validar se a interpretação de agilidade feita por este indivíduo é tão somente fundamentada em uma análise pessoal de suas habilidades ou se existe algum embasamento fundamentado na observação consciente e integral do processo da prática. (DE MELLO, DA SILVA, TRAVASSOS, 2014)

Como trabalho futuro, é sugerido a realização de novos estudos incorporando as práticas disponíveis no mercado que não foram citadas neste trabalho, bem como uma comparação entre elas, afim de descobrir características em comum. Além disso, sugere-se que seja mantido a dinâmica utilizada no trabalho atual, no que tange a recomendação de como utilizar as práticas, dessa forma, será possível corroborar para a disseminação de como usar as diferentes práticas ágeis disponíveis no mercado, afim de cada vez mais melhorar o processo de desenvolvimento de software.

REFERÊNCIAS BIBLIOGRÁFICAS

ABBAS, N.; GRAVELL, A.; WILLS, G. Historical roots of Agile methods: where did “Agile thinking” come from? *Agile Processes in Software Engineering and Extreme Programming*, p. 94–103, 2008.

AGILE ALLIANCE. (2022) Mapa do metrô para práticas ágeis. Disponível em: <<https://www.agilealliance.org/agile101/subway-map-to-agile-practices/>>. Acesso em: Julho/2022.

AOYAMA, M. Persona-and-scenario based requirements engineering for software embedded in digital consumer products. *Requirements Engineering, 2005.Proceedings. 13th IEEE International Conference on.*, p. 85–94, 2005.

APPELO, J. *Management 3.0: leading Agile developers, developing Agile leaders*. Pearson Education, 2010.

ARBULU, R.; BALLARD, G.; HARPER, N. Kanban in construction. *Proceedings of IGLC-11, Virginia Tech, Blacksburgh, Virginia, USA*, p. 16–17, 2003.

ASTELS, D. *Test-driven development: a practical guide*. 1. ed. Upper Saddle River, NJ: Prentice Hall PTR, 2003. 562 p.

ASTELS, David. *Test-driven development: a practical guide*. 1. ed. Upper Saddle River, NJ: Prentice Hall PTR, 2003. 562 p

BACH, J. *Exploratory Testing Explained*. . Disponível em <<http://www.satisfice.com/articles/et-article.pdf>> Acesso em: Julho/2022.

BALL, T.; KIM, J.; SIY, H. P. If your version control system could talk. *ICSE Workshop on Process Modelling and Empirical Studies of Software Engineering*, 1997.

BASSI FILHO, D. L. Experiências com desenvolvimento ágil. 2008. 141f. Dissertação (Mestrado em Ciência da Computação) - Instituto de Matemática e Estatística da Universidade de São Paulo, São Paulo, 2008.

Beck, K. 2004. Programação Extrema (XP) Explicada. 2004

BECK, K. Extreme programming explained: embrace change. Boston: AddisonWesley, 2000.

BECK, Kent; FOWLER, Martin. Planning Extreme Programming. 1. ed. Boston: Addison-Wesley, 2001. 139 p.

Boston, MA: Addison-Wesley, 2003. 265 p.

CONCEIÇÃO, J. D. SILVEIRA, S. D. (2015). Aplicação de Metodologias Ágeis para Desenvolvimento de Software: um estudo de caso na empresa Alliance Software.

DE MELLO, R. M., DA SILVA, P. C., TRAVASSOS, G. H. (2014) Agilidade em Processos de Software: Evidências Sobre Características de Agilidade e Práticas Ágeis. In: Anais do XIII Simpósio Brasileiro de Qualidade de Software. SBC, 2014. p. 151-165

FADEL, A. C., SILVEIRA, H. D. M. (2010) Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum e Lean. Monografia do Curso de Mestrado FT-027-Gestão de Projetos e Qualidade da Faculdade de Tecnologia–UNICAMP, v. 98, p. 101, 2010.

FEITOSA, D. S. Um estudo sobre o impacto do uso de desenvolvimento orientado por testes na melhoria da qualidade de software. Monografia (Bacharelado em Ciência da Computação), 2007.

FOWLER, M. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. EUA: Bookman, 2005.

GARTNER, M. ATDD by example: a practical guide to acceptance test-driven development. Boston: Addison-Wesley, 2012.

GEORGE, B.; WILLIAMS, L. An initial investigation of test-driven development in industry. 2003. Disponível em: <<https://collaboration.csc.ncsu.edu/laurie/Papers/>

GIL, A. C. Como Elaborar Projetos de Pesquisa. 4ª ed. São Paulo: Editora Atlas S.A., 2002.

GITHUB. As ferramentas que você precisa para construir o que você quer. Disponível em:<<https://github.com/features>>. Acesso em: Julho/2022.

HENDRICKSON, E. Driving development with tests: ATDD and TDD. Quality Tree Software, Inc. [http://www. qualitytree. com](http://www.qualitytree.com), p. 1–9, 2008.TDDpaperv8.pdf> Acesso em: Julho/2022.

HOST, M.; REGNELL, B.; WOHLIN, C. Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. Empirical Software Engineering, v. 5, n. 3, p. 201–214, 2000.

HUMBLE, J.; READ, C.; NORTH, D. The deployment production line. Agile Conference, 2006, p. 113–118, 2006.

JOVANOVIĆ, Irena, Software Testing Methods and Techniques. Jovanovic is with the DIV Inzenjering, 30, Belgrade, 2008

KOTLER, Philip; KELLER, Kevin Lane. Administração de Marketing. 2012.

LANDRE, E.; WESENBERG, H.; OLMHEIM, J. Agile enterprise software development using domain-driven design and test first. Companion to the 22nd ACM SIGPLAN conference on Object oriented programming systems and applications companion - OOPSLA '07, p. 983, 2007

LONGO, H. E. R., & Silva, M. P. (2014) A utilização de histórias de usuários no levantamento de requisitos ágeis para o desenvolvimento de software. International Journal of Knowledge Engineering and Management (IJKEM), v. 3, n. 6, p. 1-30, 2014.

MACHADO, M., MEDINA, S. G. (2009). SCRUM–Método Ágil: uma mudança cultural na Gestão de Projetos de Desenvolvimento de Software. Revista Científica Intraciência, Faculdade do Guarujá–UNIEESP, v. 1, n. 1, p. 58-71.

MAIDASANI, Dinesh. Software Testing, 1ª Edição. 2007. 389 p.

MARTIN, R. Iterative and incremental development (iid). C++ Report, n. Iid, 1999.

MASSOL, Vincent; TAHCHIEV, Petar; LEME, Felipe; GREGORY, Gary. Component-Based Software Engineering: Putting the Pieces Together. 2. ed. Stamford: Manning Publications, 2010. ISBN 1935182021.

MATTIOLI, F. E. R et al. Uma proposta para o desenvolvimento ágil de ambientes virtuais. SBC. Anais do WRVA, 2009.

MAURER, F.; MARTEL, S. Extreme programming: Rapid development for Webbased applications. IEEE Internet computing, v. 6, n. 1, p. 86–90, 2002.

MIRANDA, E. Improving Subjective Estimates Using Paired Comparisons. Software, IEEE, v. 18, n. 1, p. 87–91, 2001.

NETO, Paulo. Implementação de um módulo para integração da Abordagem use-case 2.0 com um sistema de gerenciamento de Projetos. Florianópolis, 2017. Trabalho de Conclusão de Curso (Ciências da Computação) - Universidade Federal de Santa Catarina, Florianópolis , 2017.

PATTON, J. It's All in How You Slice It. Better Software Magazine, January, 2005.

PATTON, J. User Story Mapping, 2008. Disponível em: <http://www.cs.northwestern.edu/academics/courses/394/slides/spr12/Patton_Building_Better_Products_Using.pdf>. Acesso em: Julho/2022.

PONTES, T. B., ARTHAUD, D. D. B. (2018). Metodologias ágeis para o desenvolvimento de softwares. *Ciência E Sustentabilidade*, v. 4, n. 2, p. 173-213, 2018.

POPPENDIECK, Mary; POPPENDIECK, Tom. *Lean software development: an agile toolkit*. 1. ed. Upper Saddle River, NJ: Addison-Wesley, 2003

SCHOEFFEL, P. (2021). XP Enigma-Um Jogo Educacional Não Digital para Apoio ao Ensino de Métodos Ágeis: Uma análise temporal da motivação e aprendizagem. In: *Anais do Simpósio Brasileiro de Educação em Computação*. SBC, 2021. p. 152-161.

SCHWABER, K.; SUTHERLAND, J. *Guia do Scrum*, 2009. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Portuguese-European.pdf> Acesso em: Julho/2022.

SHARP, H.; ROBINSON, H.; PETRE, M. The role of physical artefacts in agile software development: Two complementary perspectives. *Interacting with Computers*, v. 21, n. 1-2, p. 108–116, jan. 2009.

Souza, B. L. Metodologias ágeis: análise e comparação do Scrum, Kanban e Lean aplicados ao desenvolvimento de software.2021. Disponível em: https://app.uff.br/riuff/bitstream/handle/1/24054/Vers%c3%a3o%20Final_%20Beatriz%20Lopes%20Souza.pdf?sequence=1&isAllowed=y Acesso em: Julho/2022.

SUTHERLAND, J.; VIKTOROV, A. Distributed scrum: Agile project management with outsourced development teams. *System Sciences*, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, 2007.

TAVARES, H. et al. A tool stack for implementing Behaviour-Driven Development in Python Language. *arXiv preprint arXiv:1007.1722*, p. 279–284, 2010.

TELES, V., M. *Extreme Programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade*. São Paulo: Novatec, 2006.

TON, H. A Strategy for Balancing Business Value and Story Size. *Agile 2007 (Agile 2007)*, p. 279–284, ago. 2007.

WAKE, B., INVEST in Good Stories, and SMART Tasks. XP123, 2003. Disponível em <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/> Acesso em: Julho/2022.

WAZLAWICK, R. S. *Engenharia de Software: Conceitos e Práticas*. 1. ed. Rio de Janeiro: Elsevier, 2013.

WILLIAMS, Laurie; KESSLER, Robert. *Pair programming illuminated*. 1. ed.