



<https://doi.org/10.47820/recima21.v3i10.2121>

METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE ÁGIL

SOFTWARE DEVELOPMENT METHODOLOGIES AGILE

Leonardo Vedovelli¹
Renata Mirella Farina²

RESUMO: Objetivo do trabalho é apresentar mediante ao estudo de caso sobre as metodologias de desenvolvimento de software ágeis os demais benefícios apontados através da implementação do mesmo em uma organização desenvolvedora de software. Devido as novas características de desenvolvimento de software, tem-se ocasionado a procura por este segmento por grande parte das organizações. Com a entrada de um novo ciclo, com rápidas mudanças na área, tem provocado a procura ainda mais por inovação contínua, e para que isso ocorra, adotasse de métodos ágeis, até mesmo para que haja a entrega em um rápido prazo e com uma qualidade superior aos demais concorrentes, tornando assim as metodologias ágeis uma grande oportunidade de elevar inúmeros recursos dentro de uma organização.

PALAVRA-CHAVE: Organização, Desenvolvimento de Software, Métodos Ágeis.

ABSTRACT: The objective of the work is to present to the case study about software development methodologies, the other benefits pointed out through its implementation in a software development organization. Due to the new characteristics of software development, this segment has been sought after by most associations. With the entry of a new cycle, with rapid change in the área, it has provoked an even more search for continuous innovation, and for this to happen, adopting agile methods, even so that there is delivery in a fast time and with quality superior to other competitors, thus making agile methodologies a great opportunity to raise numerous resources within an organization.

KEY-WORDS: Organization, Software Development, Agile Methods.

¹ Graduando do Curso de Sistema de Informação da Universidade de Araraquara- UNIARA. Araraquara-SP. Email: vedovellileonardo@mail.com

² Orientador. Docente Curso de Sistema de Informação da Universidade de Araraquara- UNIARA. AraraquaraSP. E-mail: rmfarina@uniara.edu.br

1 INTRODUÇÃO

A nova tendência de desenvolvimento de software tem despertado grande interesse de empresas globais de desenvolvimento. Estamos entrando em uma nova era. As rápidas mudanças na tecnologia da informação, tem ocasionado na pressão da inovação contínua entre as empresas, a competição acirrada e a grande vitalidade no ambiente de negócios têm elevado ainda mais a procura pelo desenvolvimento ágil de aplicações (BOEHM, 2006).

Em vista também por altos lucros e capacidade, as organizações desenvolvedoras de software estão a buscas por novas metodologias em que possam gerenciar melhor o seu tempo e seus recursos, para a entrega de aplicações com qualidades elevadas e estimativas de prazo menores possíveis. As metodologias ágeis oferecem oportunidades para cada organização de serem líderes empresarialmente, por questões de serem notavelmente diferentes das outras metodologias tradicionais de desenvolvimento de software (QURESHI; KASHIF, 2017).

Para que as determinadas organizações entreguem as aplicações a expectativa esperada pelos seus clientes e no tempo acordante, a metodologia Ágil entra em cena como uma novidade, onde ela traz o benefício para o time, pois o caminho de desenvolvimento está profundamente organizado, desenvolvendo um software com o baixo recurso desperdiçados.

Tanto o Scrum quanto o Kanban desencadearam mudanças culturais na forma de se pensar, comportamentos e formas de tirar todos da zona de conforto e, dependendo da metodologia da empresa, geralmente acabam sendo rejeitadas pelos envolvidos, nesse caso, o objetivo deste artigo é apresentar essas ferramentas de processo ágil que utilizam práticas iterativas e incrementais para gerenciar e monitorar o desenvolvimento de uma determinada aplicação de software.

O objetivo do trabalho é apresentar mediante a revisão bibliográfica as metodologias de desenvolvimento de software ágeis contendo seus demais benefícios, através da implementação do mesmo em uma organização desenvolvedora de software e através do estudo de caso exibir o que cada metodologia ágil vem a propor em cada fase do projeto, seja ele seu início, meio e fim, correspondendo a cada segmento do setor de desenvolvimento ou gerenciamento e monitoramento do projeto.

2 REVISÃO BIBLIOGRÁFICA

Agilidade se refere à capacidade de criar e responder às mudanças nos resultados financeiros do projeto em um ambiente de negócios intenso. Agilidade é a habilidade de encontrar um equilíbrio entre flexibilidade e estabilidade (HIGHSMITH, 2004). Estrutura e organização excessivas reduzirão a criatividade e a flexibilidade e não podem suportar a tal mudanças. Pelo contrário, leva a ineficiência e leva a esforços desnecessários. (HIGHSMITH,

2004) em particular, a falta de estrutura ou estabilidade causará transtornos severos, mas estrutura demais produzirá rigidez, os métodos, práticas e técnicas de desenvolvimento ágil de projetos podem garantir o aumento da satisfação do cliente (BOEHM, 2003) para se produzir software de alta qualidade e acelerar os prazos de desenvolvimento de projetos (ANDERSON, 2003).

Trigo e Barreto (2019), destacam-se que o desenvolvimento de software tem como objetivo de atender aos requisitos do cliente dispondo do modelo cascata, também conhecido como water fall ou top-down, que é formado por etapas desenvolvidas de modo contínuo, assim sendo considerado um modelo linear, pois cada fase só pode ser realmente inicializada se a finalização da etapa anterior estiver concluída, sendo assim, uma saída é entrada para outra.

Em um ambiente de negócios globalizado, os clientes precisam obter rapidamente um retorno sobre o capital que foi investido. Ao contrário de outros métodos tradicionais, o “Scrum” priorizará claramente o retorno do investimento. A principal função do product owner é maximizar e priorizar as atualizações do backlog do produto para que os itens mais valiosos para os clientes possam ser implementados primeiro. Dessa forma, o incremento de produto obtido ao final de cada sprint retornará ao cliente várias vezes ao longo do projeto, o manifesto ágil acredita que responder as mudanças é mais importante do que seguir um plano. Por ser uma ferramenta ágil, o “Scrum” trata as mudanças como uma parte natural do processo de desenvolvimento. Conforme a lista de afazeres de produtos é atualizada – backlog, novas solicitações de clientes podem ser inseridas no sprint seguinte, o que trará para a empresa uma grande vantagem competitiva.

Ken Schwaber (2007) mencionou em seu livro “Enterprise and Scrum” que em um projeto normal, cerca de 50% do tempo é gasto nas necessidades, arquiteturas e especificações discutidas entre o gerente de projetos e o cliente, e todos esses princípios são feitos antes de qualquer projeto a ser desenvolvido, junto as funcionalidades da aplicação. No entanto, 35% dos requisitos mudam durante o projeto, e 65% das funções descritas pelos requisitos nunca ou raramente serão usadas. No modelo clássico em cascata, o cliente receberá seu produto para o seu devido uso, imediatamente após a aplicação passar por todos os testes de aceitação, implementação e testes a serem realizados, sendo assim o valor do investimento só é obtido no final do projeto.

Segundo autor Schwaber (2004), a metodologia “Scrum” não é um processo previsível e não define o que fazer em todas as situações que surgem. Geralmente é utilizado em tarefas complexas onde é impossível prever eventos do início ao fim do projeto, e fornece uma ferramenta e um conjunto de práticas que podem tornar tudo mais claro e real. Isso permite que a tenha um entendimento preciso dos fatos de todo o projeto, podendo fazer as alterações finais dentro de um determinado período, e fazer os ajustes evidentes quando necessário para atingir seus objetivos. Esta é uma das vantagens do método ágil “Scrum”: “Adaptabilidade e Flexibilidade”. Porém, entretanto a metodologia “Scrum” não é nenhum tipo de “Santo Milagreiro” e muito menos irá oferecer uma “receita pronta” para solucionar todos os seus

problemas. A única e exclusiva certeza é que esses problemas serão identificados com mais facilidade, podendo ser resolvido antes mesmo da entrega do produto final. Por se tratar de uma ferramenta, servirá como um guia de boas práticas para alcançar seus objetivos. Segundo Barboza et al. (2016, p.4): “[...] a proposta deste tipo de metodologia consiste também em particionar o desenvolvimento de software em inúmeras iterações de ciclos, que se estende em poucas semanas, em que o cliente irá receber ao final de cada ciclo uma aplicação que incluirá valor ao negócio estabelecido”.

Para Fonseca (2009), o método “Scrum” é composto por um determinado número de membros e que funciona como uma equipe altamente integrada, cada membro da equipe desempenha uma função claramente definida, e toda a equipe tem como foco o único e principal objetivo de fornecer o produto. O sucesso do método Scrum segundo Sutherland e Schawaber (2017) está ligado diretamente as pessoas, pois como o método atua em projetos e desenvolvimento de aplicações que em inúmeras vezes são difíceis de prever acontecimentos, é necessário o comprometimento das pessoas atuantes. Elimina todas as práticas de controle desnecessárias, inapropriadas e burocráticas, focando na essência do processo de desenvolvimento de aplicativos para uma determinada organização. Além do mais o trabalho em equipe é muito objetivo quando se trabalhado com a metodologia ágil “Scrum”, as metas são claras, a equipe é bem definida, flexível, comprometida e cooperativa com o projeto. E sua curva de aprendizado é relativamente baixa. Embora “Scrum” seja um método na área de exatas, o mesmo poderá ser aplicado na área de humanas, e isso é o que precisa ser considerado o tempo todo.

Confirmando essa ideia Silva, Medeiros e Medeiros (2018) destacam que existem papéis bem claros e definidos, onde cada papel possui sua necessidade exclusiva de iteração e processos que devem ser límpidos o suficiente para que cada pessoa atuante saiba exatamente o que foi, está ou será feito no projeto.

SCRUM TEAM

O time Scrum é responsável por mover todos os cards do Sprint atual e convertê-los em software para entrega. Consiste em uma equipe de 6 membros, sendo 4 desenvolvedores e 2 analistas de software para teste e que trabalham de forma participativa. No entanto, as funções não são necessariamente divididas por funções designadas, por exemplo: desenvolvedores para desenvolvimento, analistas de software para teste e assim por diante. Por se tratar de uma equipe multifuncional, caso algum analista de software saia do projeto, o teste e a qualidade do software fica sendo de responsabilidade do desenvolvedor, e assim por diante.

Em um time Scrum, é definido geralmente por 5 a 9 membros, contendo relatos de projetos Scrum sendo compostas por equipes maiores. O time Scrum é responsável por ficar inteiramente focado nas tarefas atribuídas a cards daquele determinado Sprint, e quando se

deparam com itens que impedem o progresso para término daquele card, identificam qual é o possível problema, e relatam ao “Gerente de desenvolvimento ou Gerente Geral da organização”.

Outra evidencia importante é que os próprios membros do time se gerenciam, sendo responsáveis por medir as tarefas do desenvolvimento do Sprint. Segundo FONSECA (2009), o time desenvolve de forma didática, efetuando projeto, codificação teste de unidade, aceitação e até documentação, para cada card referente ao Backlog (“Requisito”) antes de passar para o próximo Sprint, tem que ser desenvolvido todos os itens contidos no card transferido do backlog de forma minuciosamente em sistema de pilha, do que tem mais irrelevância para o menor, reforçando diariamente a formação para que cada um dos membros sejam capazes de fazer qualquer item do card a ser atribuído.

SCRUM MASTER

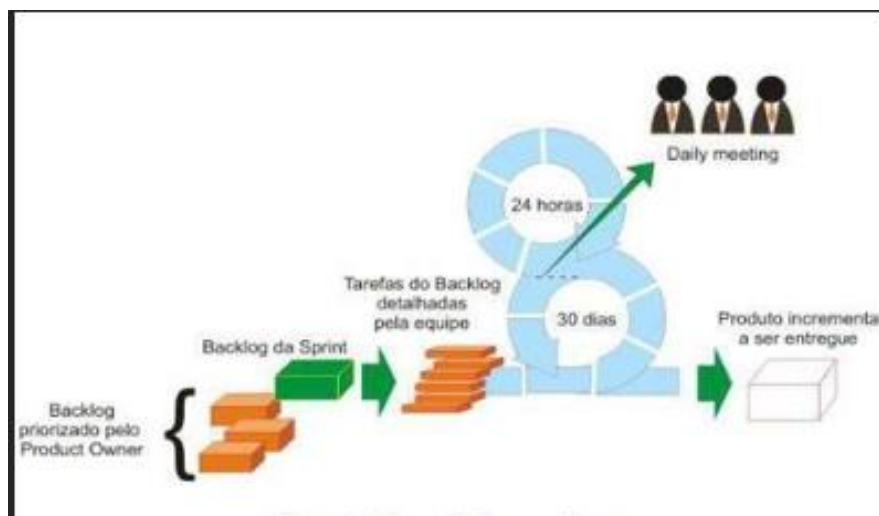
É o líder com a responsabilidade em fazer o ambiente Scrum funcionar. É ele quem irá formar o time e garantir que o mesmo esteja totalmente funcional e produtivo para as fases do Sprint, após a definição dos cards convertidos do backlog. Além das outras características atribuídas ao líder, é ele quem faz a proteção de todo o processo, afirmando que as práticas do Scrum sejam utilizadas com a disciplina necessária e assegura que o projeto e a regra organizacional estejam alinhados para que as metas do cards, estejam sendo alcançadas a cada Sprint completado (FONSECA, 2009).

Defende a integridade da equipe de interferências externas, reforça que os Sprints que iram se tornar atuais para a execução do time, não contenham itens além do que realmente pode ser entregue e executa um papel de gerenciamento do tipo “Gerente Geral de uma determinada organização” ao lado dos membros do time. Acompanha o status do trabalho diariamente realizado reuniões diárias para verificar o desempenho de cada membro do time sobre seus cards, revisão do Sprint e planejamento.

O Scrum Master é um intermediário, alguém que tem a função de fazer o time funcionar e inserir corretamente o Scrum. Pode ser interpretado como o “Gerente de Desenvolvimento” ou como dito acima “Gerente Geral de uma determinada organização”, e responde pelo BackLog em geral.

SCRUM – FLUXO DE PROCESSO

De acordo com a pesquisa de Fonseca (2009), o Scrum é dividido em um projeto em interações, e o trabalho máximo executado no Sprint atual é de duas a quatro semanas. Segue a Figura 1 descrevendo o processo.



Fonte: Adaptado de The Scrum Development Process.

Figura 1: Fluxo de Processo Scrum.

SCRUM – CICLO DE VIDA

O Scrum inicialmente foi lançado em 1993 por Jeff Sutherland em conjunto com Ken Schwaber e o nome Scrum é uma analogia a um estudo realizado em 1986 por Takeuchi e Nonaka, que teve sua publicação na Harvard Business com o intuito de mudar a forma como o desenvolvimento de software era gerenciado.

Para Koscianski (2006), o ciclo de vida útil de um determinado Scrum é particionado em três fases:

- **PRÉ-PLANEJAMENTO:** as funcionalidades são determinadas em um documento chamado BackLog. A seguir as funcionalidades são classificadas por uma ordem de prioridade, na qual são medidos “os esforços” para o seu real desenvolvimento. Nessa fase de planejamento, inclui a seleção dos integrantes que irá conter na equipe, identificação da necessidade de passar por um treinamento, as ferramentas necessárias a ser utilizada, como também uma lista com os possíveis riscos que o projeto poderá enfrentar. A fase é concluída quando se tem uma proposta de arquitetura de software. As alterações futuras deveram ser anexadas no backlog.
- **DESENVOLVIMENTO:** os riscos previamente identificados deveram ser mapeados e ter um acompanhamento bem especifica ao longo do desenvolvimento do projeto, para assim avaliar o seu impacto sobre o mesmo. Nessa fase, a aplicação é desenvolvida em determinados ciclos interativos, chamados de

“Sprint”, onde são incrementadas novas funcionalidades. Cada um desses “Sprints” dura de duas a quatro semanas de desenvolvimento tradicional, incluindo (análise, design, implementação e teste).

- **PÓS-PLANEJAMENTO:** Nessa fase irá ocorrer a integração da aplicação, elaboração dos testes finais e a documentação que o QA terá que realizar para o usuário final. A equipe se reúne para observar o estado do projeto e aplicação atual é apresentada ao cliente.

O Scrum tem cinco propriedades fundamentais, que serão descritas em Desenvolvimento:

3 DESENVOLVIMENTO

PRODUCT BACKLOG: refere-se a uma lista contendo todos os requisitos ordenados por prioridades seguindo o valor do negócio. A prioridade de um item no backlog pode sofrer alterações, requisitos podem ser incrementados ou removidos dependendo da necessidade do cliente final.

SELECTED PRODUCT BACKLOG: é o que resulta do Sprint Planning. Define o que a equipe concordou durante o planejamento do Sprint, sendo fixo (não podendo ser modificado, após a definição) durante toda a execução do Sprint.

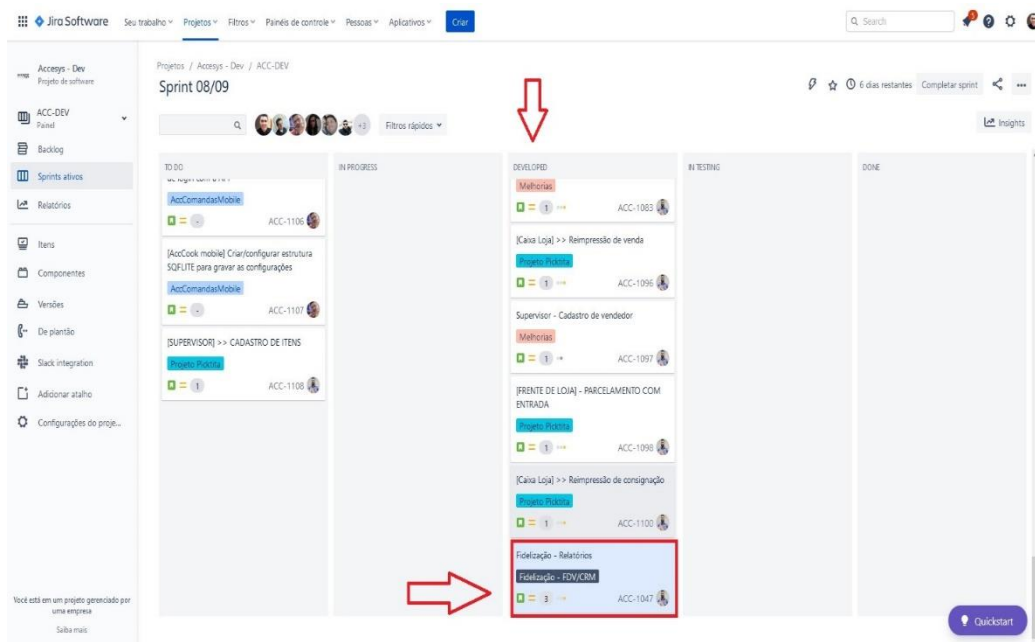
SPRINT BACKLOG: é uma lista de execuções, contendo suas respectivas estimativas de tempo (duração) do atual momento, até o final do Sprint. O mesmo é derivado a partir do Product Backlog onde são detalhados os itens do product Backlog em tarefas. É definida pela equipe, e expostas com o Product Owner. É de suma importância que a equipe tenha a determinada lista sempre atualizada.

Após realizar a classificação de todos os requisitos ordenados por prioridade, o *Product Owner* insere todos os cards contendo os requisitos prioritários ao *Sprint Backlog*. O card segue automaticamente a coluna de To Do, ficando em pendência até que o desenvolvedor responsável definido pelo *Scrum Master* atribua a coluna *In Progress*

Em detalhes do card, será apresentado todos os requisitos nos quais o desenvolvedor responsável terá que desenvolver sobre a aplicação, contendo também informações sobre o desenvolvedor responsável, o relator que sempre será o profissional de QA, Epic Link que indicará o modulo do sistema para qual será a aplicação e a categoria especificando qual tipo de desenvolvimento será aplicado.

Após o desenvolvedor ser notificado e ter conhecimento sobre o card a ser desenvolvido, ele irá atribuir a coluna In Progress, indicando que o card está em processo de desenvolvimento, caso necessite de um desenvolvimento imediato de outro card seguindo com outros requisitos mais urgentes, o desenvolvedor será autorizado a voltar esse card que está trabalhando atualmente para To Do e realizar a transferência desse outro card em caráter de urgência.

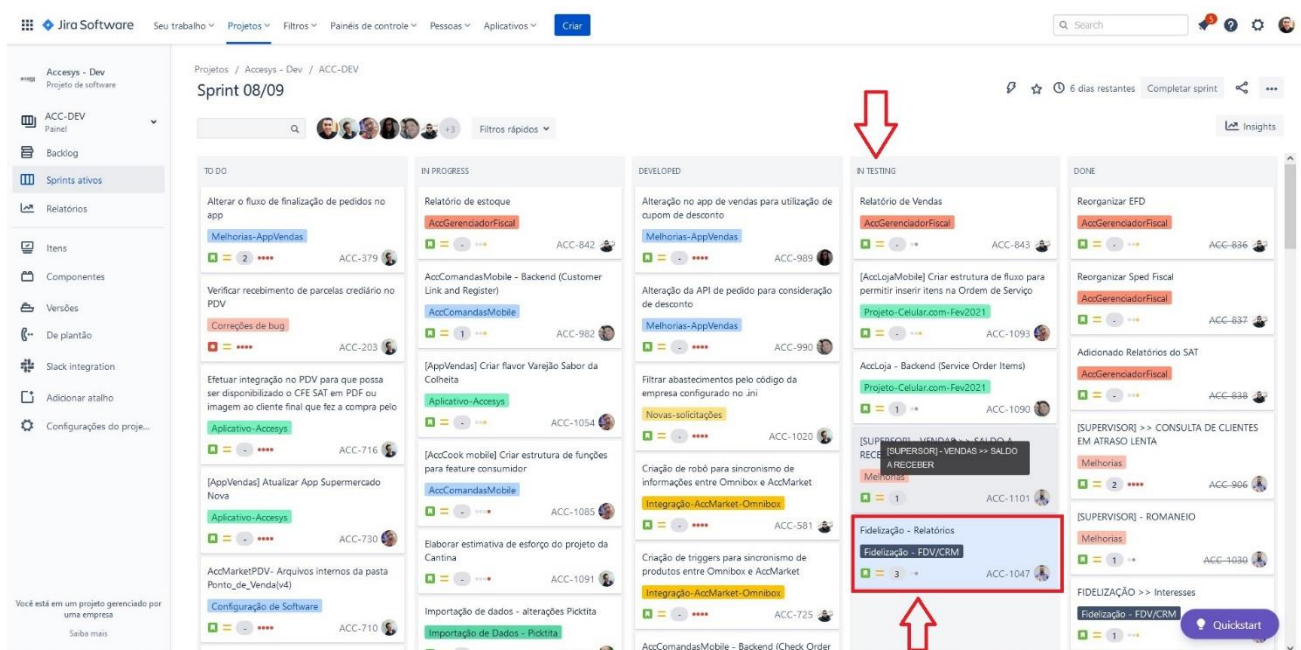
Ao término do desenvolvimento dos determinados requisitos, o desenvolvedor terá que obrigatoriamente transferir o card da coluna In Progress para coluna Developed, e será de total responsabilidade agora do profissional de QA analisar o desenvolvimento do card e realizar os determinados testes sobre o desenvolvimento. Figura 2 apresentando o término do desenvolvimento e transferência de card.



Fonte: Autoria Própria.

Figura 2: Término de desenvolvimento sobre o card e transferência para coluna Developed.

Fase final sobre o desenvolvimento dos requisitos, aonde o profissional de QA irá transferir o card para In Testing, significando que aquele determinado card está sendo analisado e testado pelo departamento de garantia de qualidade. Se localizado algum tipo de problema, ou se surgir observações sobre a aplicação, será de responsabilidade do profissional de QA, realizar as devidas anotações sobre o card e obrigatoriamente voltar o card para a coluna de To Do, notificando o desenvolvedor sobre card. Figura 3 apresentando transferência de card pelo departamento de QA e realizando anotações em card sobre o teste realizado sobre ele.

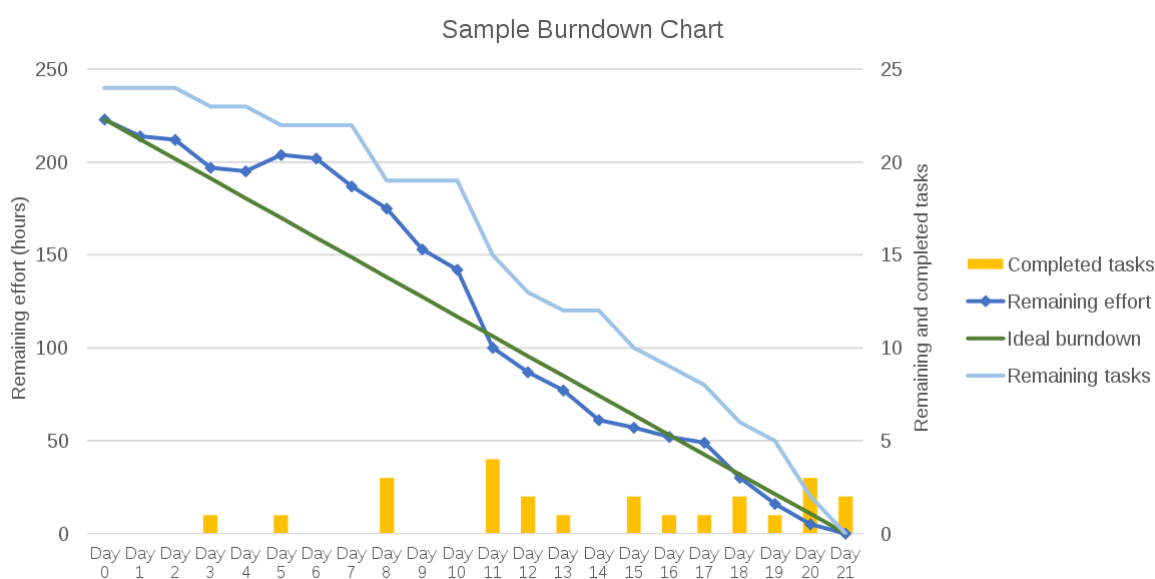


Fonte: Autoria Própria.

Figura 3: Transferência de card feito pelo setor de QA para coluna In Testing.

Caso o departamento de QA realizar a aprovação do desenvolvimento, o card será obrigatoriamente transferido para coluna Done, representando assim para todos os membros do Sprint, que aquele desenvolvimento está pronto para ser lançado em ambiente de produção, aguardando assim o final do Sprint para geração do pacote de atualização e encaminhado ao setor de análise de TI para ser implantado em cliente final. Assim se repetindo todo ciclo sobre os outros card's contidos no Sprint atual com a sua duração.

BURNDOWN CHART: é um gráfico que exibe a quantidade de trabalho que se acumularam ao restante de um Sprint, seguindo dia por dia. Neste gráfico a altura irá indicar a quantidade de tarefas do Sprint Backlog que não foram completadas, e o comprimento determinam os dias. Este gráfico é considerado um dos principais recursos de medição em todo o processo de desenvolvimento, sendo também um diferencial na parte de metodologia ágil Scrum. Quanto mais horizontal, melhor. Figura 4 representa o gráfico.



Fonte: Autoria Própria.

Figura 4: Gráfico sobre a quantidade de atividades acumuladas.

IMPEDIMENT BACKLOG: é a lista contendo todos os problemas que atrapalham a equipe de evoluir sobre o Sprint que está atualmente trabalhando. Tais problemas devem ser resolvidos pelo Scrum Master. Pode ser dividida em duas outras listas: Team Impediment, que são os impedimentos que podem ser resolvidos pela própria equipe, e Organization Impediment, na qual é a lista que a equipe não pode resolver.

FERRAMENTA – JIRA SOFTWARE

Jira é um software destinado a realização de gerenciamento e monitoramento de todos os Products Backlog criados, permitindo a seleção de determinados card's que serão inseridos para o Sprint que está sendo planejado, organiza-se todas as tarefas entre as colunas inseridas e que por sua vez terá seus card's posicionados de acordo com cada etapa em que está sendo desenvolvido, contendo também os dias restantes para conclusão daquele Sprint atual. Cada card tem seu próprio desenvolvedor que por sua vez é responsável em direcionar o card para a coluna na qual está realizando ou termino do procedimento.

ESTUDO DE CASO APLICANDO AS METODOLOGIAS XP E SCRUM

Um estudo de caso foi executado em um time de desenvolvimento com 15 colaboradores, que se formou para trabalhar em um projeto de software para gestão de ensino, em uma determinada organização de software que já desenvolvia outros produtos.

Após passar por diversas reuniões e estudos de mercado, o time chegou a uma conclusão de que seria necessário realizar alterações a suas atitudes em relação ao desenvolvimento de software, pois atualmente o mercado oferece novos conceitos e novas necessidades. Conforme análise feita sobre o paradigma atual, obteve-se a adoção de práticas ágeis, pois este seria o primeiro projeto de desenvolvimento ágil aonde os requisitos não se possuíam com precisão.

METODOLOGIA ÁGEIS APLICADAS NO PROJETO

A figura 5 exibe a junção entre as práticas selecionadas entre as metodologias ágeis XP e SCRUM para serem aplicadas no projeto de gestão de ensino. Particionado as práticas por metodologia e foco, abordando o nome pela qual a prática é geralmente conhecida, a metodologia que utiliza a mesma, uma descrição de sua aplicação dentro desse estudo de caso e a alegação que permitiu a escolha das práticas para o projeto.

PRÁTICA	ORIGEM	DESCRIÇÃO	JUSTIFICATIVA
Práticas Focadas no Desenvolvimento			
Código Coletivo	XP	Código fonte disponível para qualquer desenvolvedor a qualquer momento.	Tornou-se necessário adotar essa prática devido à necessidade da implementação ser feita em colaboração, ou seja, vários desenvolvedores no mesmo projeto.
Refatoração	XP	A refatoração está presente no projeto devido ao código coletivo, quando um desenvolvedor visualiza uma possibilidade de melhoria, isto é feito e documentado no próprio código, criando o histórico dessas alterações.	A coletividade do código também gera responsabilidades a todos os desenvolvedores de perceberem possibilidades de melhoria.
Simplicidade	XP	Todo o código produzido é elaborado com o máximo de simplicidade para que possa ser economizado tempo de produção e entendimento da estrutura do código.	A utilização da simplicidade justifica-se pelo fato do código ser coletivo; ou seja, todos precisam entender o que foi implementado.
Foco na Codificação	XP	A documentação tem a função de orientar o desenvolvedor as necessidades do cliente, porém não é o foco da equipe de analistas desenvolver documentação.	A prioridade da equipe é software rodando com base nos princípios ágeis; porém, a documentação não é dispensada.
Integração Contínua	XP	Toda a produção do projeto é organizada no servidor automaticamente pela ferramentas utilizadas, o código produzido pelos desenvolvedores, documentação pelos analistas, e requisitos pelos clientes e gerentes.	O desenvolvimento em colaboração exige que tudo esteja centrado em um único lugar permitindo assim que essas informações sejam repassadas a todos da equipe.
Testes	XP	Após o término de cada <i>sprint</i> semanal ou quinzenal é repassado o código compilado em executáveis para o departamento de testes, após a conclusão dos testes os desenvolvedores trabalham em rodízio para correção de <i>bugs</i> .	Para evitar a entrega de produtos com um número elevado de erros para o cliente faz-se necessário estabelecer uma rotina de testes.
Programação Pareada	XP	O desenvolvimento em pares ocorre quando uma funcionalidade requer conhecimento de vários desenvolvedores; sendo assim, estabelece um par de desenvolvedores responsável para implementar essa funcionalidade.	A necessidade de desenvolver em pares ocorre quando o domínio sobre o problema não é total por somente uma pessoa, faz-se assim a junção de pares para a proposta do desenvolvimento em duplas.
Práticas Focadas no Gerenciamento			
Product Backlog	SCRUM	Volume de documentação inicial do projeto que foi extraída na fase de análise de requisitos pelos analistas contém as informações sobre o escopo do projeto, ficando no portal do projeto e, é acessível por todos os membros da equipe.	Essa documentação engloba a estrutura fundamental do projeto é composta pela lista de todos os módulos do sistema com a descrição de suas funcionalidades que foram estabelecidas juntamente com o cliente.

<i>Sprints</i>	SCRUM	As iterações têm duração de duas semanas, que é a divisão do desenvolvimento, que por sua vez, compõem a lista de funcionalidades a serem desenvolvidas.	Os <i>sprints</i> são como o plano a ser seguido o mapa para a equipe possa se orientar no que desenvolver durante o prazo do ciclo.
<i>Sprints Backlog</i>	SCRUM	No planejamento de cada <i>sprint</i> retiram-se as funcionalidades a serem desenvolvidas durante aquela fase; é repassado para o papel e exposta no quadro atual da equipe para que seja fácil o acesso e visível a todos.	Durante cada <i>sprint</i> é feita à lista dessas funcionalidades a serem desenvolvidas para que seja possível acompanhar as tarefas da equipe.
<i>Sprint Retrospective</i>	SCRUM	Quando fecha-se uma <i>sprint</i> é feita uma retrospectiva para que os problemas encontrados pela equipe sejam discutidos, isto para não ocorrer no futuro, e também os acertos para servirem como base a todos.	Essa prática torna-se fundamental à medida que o projeto é desenvolvido, para que a equipe possa crescer com os erros e acertos.
<i>ScrumMaster</i>	SCRUM	O gerente do projeto, líder da equipe, tem papel fundamental na parte gerencial do projeto, auxilia toda a equipe, resolve problemas que possa estar prejudicando o andamento das tarefas, exige dos colaboradores, delega responsabilidades, marca reuniões e em contato direto com os patrocinadores.	A liderança máxima na equipe em relação aos participantes da linha de produção é de fundamental importância, pois o seu papel principal é facilitar o trabalho dos membros da equipe.
<i>Daily Scrum Meeting</i>	SCRUM	Pequenas reuniões diárias com a participação de todos os integrantes da equipe, para que possa ser dado sempre o status do projeto, e também para que o <i>ScrumMaster</i> possa resolver qualquer problema que impeça a equipe de desenvolver o <i>sprint</i> dentro do tempo hábil.	As reuniões são de grande importância para o projeto, pois é onde as informações sobre o status atual do projeto é dada para todos, também o líder <i>ScrumMaster</i> exerce o seu papel de facilitador das tarefas da equipe.

Fonte: Análise de Gerenciamento de Projeto de Software Utilizando Metodologia Ágil XP e Scrum: Um Estudo de Caso Prático.

Figura 5: Práticas ágeis das metodologias XP e Scrum inseridas no estudo de caso.

Nota-se a diferença de centralização entre as metodologias, enquanto a XP é direcionada para as práticas de implementação de código, a SCRUM é direcionada ao gerenciamento e planejamento, como também a parte de estratégia do projeto.

Em figura 6 somasse as melhores práticas levantadas neste estudo de caso, porém, algumas praticas que constam nas metodologias não poderiam ser destacadas perante a estratégia adotada pela empresa que emprega o time de desenvolvimento.

PRÁTICA	ORIGEM	JUSTIFICATIVA	IMPORTÂNCIA
Cliente sempre presente	XP	Apesar de existir consultores o projeto foi iniciado sem patrocinador específico o que impediu a presença constante desse ator de papel fundamental no início do projeto.	Com o cliente sempre presente é possível ter sempre em mãos os requisitos diretos da fonte; ficando mais simples para a equipe entender os requisitos propostas.
Semana de 40 horas	XP	Os membros da equipe do projeto também atuam em outros projetos desenvolvidos pela empresa que acolhe a equipe; portanto, os desenvolvedores têm semanas de 44 horas trabalhando em expedientes de segunda a sexta-feira de 8 horas e aos sábados de 4 horas.	Trabalhar com colaboradores sempre motivados e descansados é um dos princípios do manifesto ágil, permite extrair do colaborador o máximo do seu poder criativo, fundamental para desenvolver-se software.

Fonte: Análise de Gerenciamento de Projeto de Software Utilizando Metodologia Ágil XP e Scrum: Um Estudo de Caso Prático.

Figura 6: Melhores práticas inseridas no estudo de caso.

4 RESULTADOS

As metodologias ágeis impostas em um desenvolvimento de software rompem o paradigma do desenvolvimento em cascata, e outros procedimento realizados que são considerados tradicionais, porém elas não são de fato uma substituição aos procedimentos já atualmente existentes, mais sim um apêndice ou uma outra possibilidade real.

A uma nova iteração um subprojeto é discriminado, onde são propostas todas as etapas tais como: Planejamento, requisitos, codificação e testes. Essas iterações duram poucas semanas, o que caminha a resultados rápidos, principalmente para os investidores “Clientes finais”. Os subprojetos entregues regularmente aderem a funcionalidades já em operação.

Os times de desenvolvimento ágeis frequentemente são menores, tornando a comunicação simples e segura, a qual diariamente ocorre em tempo real, o que leva a um resultado com baixas documentação do projeto, considerando a um ponto adversário.

O cliente final se torna parte da equipe de desenvolvimento, propondo sugestões e melhorias, possui presença no planejamento do escopo e de cada iteração do projeto ou subprojeto, aprovando cada entrega realizado pelo time de desenvolvimento. O projeto se mantém com grandes fanhos com a atuação do cliente, pois as mudanças podem acontecer logo no início de cada fase sem afetar a qualidade e os custos do desenvolvimento.

A metodologia ágil SCRUM foi estabelecida inicialmente para o monitoramento e gerenciamento de projetos e seus subprojetos. Tendo em vista que o XP é geralmente utilizado durante o desenvolvimento das determinadas aplicações, devido a execução de suas técnicas e ferramentas. A metodologia ágil Lean acaba não se tornando uma prática utilizada em desenvolvimento ou em um monitoramento e gerenciamento de projetos, mas atua como um conjunto de teorias, atributos e

ferramentas que transforma o desenvolvimento enxuto. Com esse estudo, pode-se garantir que XP, SCRUM e Lean podem ser utilizados como metodologias complementares.

Em Resumo, Franco (2007) destaca as características frequentes entre as metodologias ágeis de tal forma:

- **TESTES:** Modo convencional que trata a implementação e os testes como fases inteiramente distintas. Em metodologia ágeis, esta divisão tende a se unificar. Implementação e testes acontecem inúmeras vezes juntos, aonde o mesmo desenvolvedor que elabora o código, tem o conhecimento e logica para se realizar os testes;
- **DESENVOLVIMENTO ITERATIVO:** O desenvolvimento é realizado entre ciclos (iterações), que tem como proposito produzir e embarcar partes do software. Cada ciclo pode se manter desde alguns meses até mesmo em poucas horas, dependendo a metodologia escolhida e as experiencias de cada membro da equipe. Sendo dessa forma, o procedimento acaba se tornando flexível para conciliar mudanças funcionais e de prioridade perante a construção da determinada aplicação. Ao fim de cada ciclo, a aplicação poderá ser entregue ao cliente final para que a sobra do desenvolvimento seja realocada pelo seu feedback.
- **DESENVOLVIMENTO INCREMENTAL:** Ao decorrer das iterações, a aplicação poderá acolher incrementos funcionais estabelecidos de duas formas: 1) adicionando funcionalidades a nível que a aplicação se expande, ou 2) transformando as funcionalidades junto com a aplicação. Na primeira, as funcionalidades são executadas inteiramente e entregues umas por vez, no segundo caso elas são elaboradas de forma reduzida para entrarem em ambiente de produção rapidamente e, se preciso, são niveladas ou aperfeiçoadas nas próximas iterações;
- **COLABORAÇÃO:** Clientes finais e usuários se aproximam dos desenvolvedores e compreendem o aperfeiçoamento da aplicação. O contato diariamente com o cliente final estabelece um feedback rápido e estreita a comunicação entre as pontes. Com mais comunicação, a aparição dos integrantes sobre o decorrer do projeto torna-se mais elaborada, evitando assim que eventuais surpresas ocorram ao fim do projeto. O reconhecimento e resolução de problemas tornam-se mais

rápido e as preferências, o escopo e os detalhes da implementação podem ser debatidos e negociados com mais propensão;

- **ESTIMATIVAS:** Metodologia ágeis utilizam aferições ao invés de prognóstico. Essas aferições são constituídas por uma dupla de valores $\langle v, p \rangle$, onde v é uma intuição sobre algum determinado evento ou atividade e p é a possibilidade de v acontecer. Os times ágeis sustentam-se na comunicação estabelecida e na transparência. Ao invés de levar suas estimativas como fato, consideram que existe uma ambiguidade relacionada ao valor estimado e provam isso para que o cliente final e outros membros também tomem ciência do nível de dificuldade de cada tarefa atribuída. Estimativas de grande prazo normalmente possuem níveis maiores de incertezas relacionadas. À medida que o tempo passa e o entendimento sobre o assunto eleva, as estimativas podem ser reclassificadas considerando um nível maior de detalhes e agregando inúmeras probabilidades de desenlace mais alto;
- **NEGOCIAÇÃO:** No desenvolvimento de uma aplicação, a organização e o produto estão relacionados entre quatro variáveis correlativas que são: tempo, custo, escopo e qualidade. Essas variáveis se juntam de forma que a alteração do valor de qualquer uma delas afetam as demais;
- **PRIORIZAÇÃO:** Metodologia ágeis se regem fortemente na adequação de mudança. As estratégias de organização focam em planos detalhados para o curto prazo e mais leve para o futuro remoto. Sendo assim, é possível um olhar de forma panorâmica para levar as decisões ao longo prazo e exatidão nas atividades do dia a dia. Existem duas vantagens consideradas principais neste contexto que são: 1) economia de esforço ao deixar de lado o detalhamento de longo prazo, pois é difícil realizar antecipações do futuro e a alta chance de modificação durante a realização nula facilmente especificação atenciosas; 2) detalhes no longo prazo poderá acarretar a falsa sensação de certeza, pois os membros poderão tratar antecipações aprofundadas como previsões. Equipes ágeis analisam seus planos continuamente. A cada organização, eles obtêm a oportunidade de estimar as condições do projeto e, fundamentada nesses fatos, escolher o melhor caminho para chegar em seus objetivos. Esta estratégia conserva os planos e a execução sempre apropriados a realidade, que rigorosamente estão em alteração, significando localizar prioridades para cada caminho do projeto.

5 CONCLUSÃO

Assim presente em toda Metodologia Ágil utiliza o princípio básico “Mudança é a única constante” – são palavras opostas neste real cenário que indicam que os requisitos irão variar ao longo do projeto de desenvolvimento, de uma forma ou de outra.

A verdadeira e única afirmação que se tem certeza é que os requisitos irão mudar, apenas não sabendo quando. Porém, a mudança relacionada ao (de escopo, nas regras de negócio, leis governamentais vinda de Municípios, Estaduais ou Federais) iram acontecer.

Todo profissional, principalmente o de tecnologia devera sempre estar preparado e atualizado. O SCRUM foca as pessoas que desenvolvem a aplicação e não no processo diretamente, que geralmente é direcionado a seguir requisitos de forma muito rígida definidos no início do projeto. Irá além das metodologias tradicionais (que tem já um certo crescimento e amadurecimento) que focam o processo (parte documental) e fazem hipótese de que os requisitos não sofreram certas alterações.

O desafio proposto futuramente as metodologias ágeis será identificar meios de minimizar as suas desvantagens encontradas sem transformá-las em metodologias pesadas, também elevando o número de pessoas que integram a equipe sem perder a confiabilidade e eficiência no gerenciamento de mudanças presentes.

Assim como outros processos e metodologias ágeis, o Scrum obtém de vantagens e desvantagem, como todos. Porém, ao analisar com certo cuidado, pode ser uma eventual saída, seja a implantação de um modelo híbrido.

Sendo assim o melhor método é aquele em que os profissionais consigam conciliar as necessidades dos projetos e das organizações associados à qualidade nas entregas, capacidade de controle de custos, prazos, riscos e dentre outros eventuais (SANTOS; SANTOS; SHIBAO, 2017).

REFERÊNCIAS

ANDERSON, D. J., *Agile Management for Software Engineering, Applying the Theory of Constraints for Business Results*, Prentice Hall, 2003.

BOEHM, B., *A View of 20th and 21st Century Software Engineering*, ICSE 2006.

BOEHM, B. and Turner, R., *Balancing Agility and Discipline A Guide for the Perplexed*, Addison Wesley, 2003.

BARBOZA, L. F. et al. Análise comparativa entre as abordagens ágil e tradicional de gestão de projetos: Um estudo de caso no setor industrial. In: Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade, 5, 2016, São Paulo. Anais... São Paulo: UNINOVE, 2016.

FONSECA, Isabella, *Engenharia de Software Conference*, São Paulo, DevMedia, 2009.

FRANCO, E. F.: Um modelo de gerenciamento de projetos baseado nas metodologias ágeis de desenvolvimento de software e nos princípios da produção enxuta. Escola Politécnica da Universidade de São Paulo (Dissertação de Mestrado). 2007.

FADEL, Aline Cristiane. SILVEIRA, Henrique da Mota, *Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum e Lean*, 2010. Disponível em https://d1wqtxts1xzle7.cloudfront.net/3307430/Lean_Agil_v8-with-cover-page-v2.pdf?Expires=1630553439&Signature=Cco3Ow~X72e~lIJ41O29Rv-zjLZbhNG8tk00WNfilb~ukOzvv4t1Im7B~A80fwZHqgVcnUX-sTKc40Nz2gYq5UrtVSgP1cPu~eh~DGjaUPZQMUKuT-LiAI4B~9m-DVDyEIFMJBloQQ2d-QoCvEvcV~CX7SkH1ssaGFR6gGo2Pc2HXiQO23-Z2Tdg4Dvu4lIXTHw9s-aFuAfpHyvrOFZyNlzMzBwx2Pal4ujr~cr0Y9OqhViPn~QQYBsnt7j90dYDVrKt1pgyt~u-8jDeoYhM6q7VF~iulDORVmO5af-8RhYn4m60fvcOdpVBzNLyiuWD6uU7vX2xntYq1iJsPvgOxw_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA Acesso em: 31 Ago.2021.

HIGHSMITH, J., *Agile Project Management, Creating innovative products*, Addison Wesley, 2004.

KOSCIANSKI, A., Soares, Santos, M. *Qualidade de Software*, São Paulo, Novatec Editora, 2006.

LINHADECÓDIGO, *Uma metodologia ágil – SCRUM*, 2021. Disponível em <http://www.linhadecodigo.com.br/artigo/2084/uma-metodologia-agil-scrum.aspx>> Acesso em: 08 abr. 2021.

MACHADO, Marcos, *SCRUM – Método Ágil: uma mudança cultural na Gestão de Projetos de Desenvolvimento de Software*, SG Medina, Revista Científica Intraciência, 2009. Disponível em < http://uniesp.edu.br/sites/_biblioteca/revistas/20170531154126.pdf> Acesso em: 25 Ago. 2021.

PRESSMAN, R., *Engenharia de Software*, São Paulo, Mc Graw-Hill, 2006.

QURESHI, M. R. J.; KASHIF, M. Adaptive framework to manage multiple teams using agile methodologies. *International Journal of Modern Education and Computer Science*, v.9, n.1, p.52-59, 2017. <https://doi.org/10.5815/ijmeecs.2017.01.06>.

SCHWABER K., *Agile Project Management With Scrum*, Microsoft Press, 2004. _____, *The Enterprise and Scrum*, Microsoft Press, 2007.

SANTOS, P. R.; SANTOS, M. R.; SHIBAO, F. Y. Comparação entre os padrões de gerenciamento de projetos PMBOK, ICB e PRINCE2. *Caderno de Administração*, v. 25, n.2, 58-73, 2017. <https://doi.org/10.4025/cadadm.v25i2.36119>.

SAVOINE, Márcia. MARTINS, Lucyano. ROCHA, Mayton. SANTOS, Cirlene, **Análise de Gerenciamento de Projeto de Software Utilizando Metodologia Ágil XP e Scrum: Um Estudo de Caso Prático**, TO Araguaína, XI Encontro de Estudantes, 2009. Disponível em <<http://ulbra-to.br/encoinfo/wp-content/uploads/2020/03/An%C3%A1lise-de-Gerenciamento-de-Projeto-de-Software-Utilizando-Metodologia-%C3%81gil-XP-e-Scrum-Um-Estudo-de-Caso-Pr%C3%A1tico.pdf>> Acesso em: 27 Ago. 2021.

SUTHERLAND, Jeff; SCHWABER, Ken. *The Scrum Guide™*. 2017. Disponível em: <<https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>> Acesso em: 09 set 2021.

SILVA, Todman Reis da; MEDEIROS, Marcus Vinicius Batella; MEDEIROS, Glaucia Rodrigues Nascimento. *Gestão de Riscos no Framework SCRUM Utilizando Análise SWOT*. *Revista de Tecnologia da Informação e Comunicação da Faculdade Estácio do Pará*. vol. 1, num. 1, pag. 48–5. Belém, Abril 2018. Disponível em: <<http://revistasfap.com/ojs3/index.php/tic/article/view/172>> Acesso em: 09 set 2021.

TRIGO, Ramon Alves; BARRETO, Luiz Claudio. *Evolução dos Métodos de Desenvolvimento de Software em Microempresas*. *Revista Gestão em Foco - Edição nº 11 – Ano: 2019*. Disponível em: <http://portal.unisepe.com.br/unifia/wp-content/uploads/sites/10001/2019/08/009_EVOLU%C3%87%C3%83O-DOS-M%C3%89TODOS-DE-DESENVOLVIMENTO-DE-SOFTWARE-EM-MICROEMPRESAS.pdf> Acesso em: 08 Set. 2021.