

**METODOLOGIA ÁGIL SCRUM PARA DESENVOLVIMENTO DE
SOFTWARE: estudo a partir de um relato de implantação**

***AGILE SCRUM METHODOLOGY FOR SOFTWARE DEVELOPMENT: Study
from an Implementation Report***

Vitor Pires Loddi – vitor_loddi@hotmail.com
Faculdade de Tecnologia de Taquaritinga (FATEC) –SP – Brasil

Ronaldo Ribeiro de Campos – ronaldo.campos@fatectq.edu.br
Faculdade de Tecnologia de Taquaritinga (FATEC) –SP – Brasil

DOI: 10.31510/infra.v18i1.1085

Data de submissão: 06/03/2021

Data do aceite: 09/07/2021

Data da publicação: 30/07/2021

RESUMO

As empresas de desenvolvimento de *software* têm buscado controlar com maior facilidade suas demandas, realizando a entrega de produtos com maior qualidade e no menor prazo possível. Nesse esforço, a ausência de um determinado padrão para ser seguido pela equipe de desenvolvimento acaba sendo um problema diante da procura de um maior domínio das necessidades no desenvolvimento de sistemas. Para propor soluções para esta dificuldade foram desenvolvidas algumas metodologias. Este trabalho se concentra no estudo dessas metodologias, particularmente aquelas conhecidas como *movimento ágil*, dando ênfase à metodologia *Scrum* e apresentando características, métodos, efeitos, benefícios bem como problemas que possam ser provenientes de sua utilização. Em termos de metodologia, este trabalho utilizou a pesquisa bibliográfica por meio de levantamento realizado em livros, artigos e outros materiais relativos ao assunto. Na revisão teórica foram apresentadas as etapas, procedimentos, reuniões e cargos necessários para a implementação do *Scrum* em uma organização. Em seguida foi escolhido, entre os casos pesquisados na literatura, um caso de implantação da metodologia e relatado suas ocorrências, detalhando as dificuldades e adaptações necessárias para a aplicação da metodologia, assim como alguns dos resultados alcançados. Como resultado do estudo, foi possível identificar a adequação da metodologia à situações práticas.

Palavras-chave: Scrum. Metodologia ágil. Sprint. Desenvolvimento.

ABSTRACT

Software development companies have sought to control their demands more easily, delivering products with the highest quality and in the shortest possible time. In this effort, the absence of a certain pattern to be followed by the team ends up being a problem in the face of the search for this new form of systems development. To propose solutions to this difficulty, some methodologies were developed. This work focuses on the study of these methodologies, particularly those known as agile movement, emphasizing the Scrum methodology and presenting characteristics, methods, effects, benefits as well as problems that may come from its use. In terms of methodology, this work used bibliographic research through a survey carried out in books, articles and other materials related to the subject. In the theoretical review, the steps, procedures, meetings and positions necessary for the implementation of Scrum in an organization were presented. Next, among only cases researched in the literature, a case of implementation of the methodology and reported its occurrences, detailing the difficulties and adaptations necessary for the application of the methodology, as well as some of the results achieved. As a result of the study, it was possible to identify the adequacy of the methodology to practical situations.

Keywords: Scrum. Agile methodology. Sprint. Software development.

1 INTRODUÇÃO

Conforme Fadel e Silveira (2010) um dos maiores impasses no setor de desenvolvimento de *software* é compreender os reais problemas e gerar resultados para a solução deles, de forma a deixar o cliente satisfeito. Encontra-se facilmente projetos concluídos em que se pode notar que as especificações do projeto são insuficientes.

Na tentativa de otimizar tempo e ampliar os lucros as empresas de *softwares* se empenham na tentativa de encontrar ferramentas que possam auxiliar em ganho de benefícios e de uma melhor qualidade para seus consumidores.

Segundo Fadel e Silveira (2010), para que as empresas desenvolvedoras atinjam as expectativas esperadas por seus usuários, muitas optam por aderir a utilização de uma metodologia ágil, a qual proporciona uma maior organização e maior eficácia para seus colaboradores e cujo uso permite ter um melhor aproveitamento dos recursos utilizados para a elaboração de um projeto.

Este estudo baseia-se no contexto das chamadas metodologias ágeis, dando foco na metodologia *Scrum*. Por meio do uso da pesquisa bibliográfica foram apresentados seus principais atributos, peculiaridades, benefícios e dificuldades no processo de implantação. Em seguida foi escolhido entre os relatos obtidos com a pesquisa bibliográfica um caso de uso do *Scrum*, para dar maior base à análise e considerações finais deste estudo.

2 METODOLOGIAS ÁGEIS

De acordo com Martins et.al (2009) devido às mudanças com facilidade em projetos de desenvolvimento de *software* era preciso criar algum procedimento capaz de tornar a aceitação a alterações em um projeto mais flexível, ocasionando a menor perda possível, desse modo os custos não teriam um acréscimo tão exponencial, já que os métodos convencionais demandam um elevado tempo em análise e planejamento.

Segundo Ambler (2004) apud Martins et.al (2009) por conta dessa necessidade em fevereiro de 2001, dezessete desenvolvedores de *software* se propuseram a estabelecer novas práticas para resolver os problemas identificados, estes profissionais de diversas áreas de formação com visões diferentes sobre os modelos e métodos de desenvolvimento de *software* em comum, criaram um manifesto para encorajar melhores meios de desenvolvimento.

Conforme Martins et.al (2009) o documento foi denominado de Manifesto Ágil ou *Agile Manifesto* e foi criado por vários desenvolvedores, mas teve como idealizadores Kent Beck, Ken Schwaber, Martin Fowler e Jim Highsmith. onde definiu-se alguns pilares e padrões a serem estabelecidos para obter o êxito no desenvolvimento ágil de *software*, Tais princípios ágeis são:

- Indivíduos e interações são mais importantes que processos e ferramentas;
- *Software* em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano; ou seja, mesmo havendo valor nos itens à direita, valoriza-se mais os itens à esquerda.

Segundo Rachid e Stopa (2019) o propósito do manifesto é satisfazer o cliente, realizando contatos frequentemente entre equipe e cliente, fazendo com que as entregas ocorram periodicamente, dessa maneira a equipe permanece motivada e trata de uma forma menos complicada referente aos problemas que possam ocorrer.

3 SCRUM

Conforme Pham (2012) apud Conceição e Silveira (2015) o termo *Scrum* surgiu em um artigo publicado por Hirotaka Takeuchi e Ikujiro Nonaka na *Harvard Business Review* de 1986.

Nesse artigo, intitulado “*The new product development game*” (O novo game do desenvolvimento de produtos) Takeuchi e Nonaka descreveram uma abordagem holística, na qual equipes de projeto são formados por equipes reduzidas multifuncionais, trabalhando com êxito em busca a um objetivo comum.

Para Prikladnicki, Willi e Milani (2014) apud Conceição e Silveira (2015) o *Scrum* é um *framework*¹ ágil que auxilia no gerenciamento de projetos complexos e na elaboração de produtos. É conhecido como um *framework* que prescreve um conjunto de práticas leves e objetivas, muito utilizadas na área de desenvolvimento de *software*. As práticas do *Scrum* também podem ser utilizadas para projetos de outra natureza, desde que possuam certo grau de complexidade, pois só assim suas práticas de inspeção e adaptação fazem sentido.

Segundo Franco (2007) apud Fadel e Silveira (2010) O *Scrum* não determina uma estratégia específica para a criação de *software* durante a etapa de execução, trata-se de uma metodologia que ressalta o gerenciamento do time, concentrado no planejamento das etapas, na maneira como cada passo deve ser realizado, permitindo que cada colaborador faça a sua escolha e se encarregue da melhor maneira para concretizar com êxito seu ciclo.

Conforme Martins et.al (2009) a aplicação da metodologia *Scrum* obteve um enorme ganho no setor de produção de *software*, isso ocorre por conta de ser uma metodologia que se adapta com facilidade, com grande praticidade e principalmente onde é possível realizar alterações e mudanças nos requisitos e funcionalidades durante o processo de desenvolvimento do *software*.

De acordo com Martins et.al (2009) O *Scrum* consegue realizar entregas mais eficazes, com maior qualidade e em menor tempo, conseguindo adaptar as etapas de acordo com o desejo do cliente e se moldando conforme as reais necessidades, utilizando como princípio os fundamentos do manifesto ágil orientando os procedimentos de concepção.

3.1 Papéis do *Scrum*

De acordo com Lovato e Silva (2016) o *Scrum* possui três principais papéis dentro da implementação dos processos de *Scrum*, cada um desses cargos possui seu dever e comprometimento para que a sua execução seja bem-sucedida. Tais cargos são:

¹ Conforme Johnson & Foote (1988) apud Exterkoetter et.al (2003) é um composto de classes que agrega um plano abstrato para solucionar a um conjunto de problemas relacionados.

Product Owner: Segundo Lovato e Silva (2016) trata-se de um profissional perito em negócios, desempenhando o papel do cliente, tendo conhecimento profundo referente às regras de negócios. O *Product Owner* terá proximidade tanto da equipe de desenvolvimento como de seus clientes, dessa maneira tendo uma compreensão maior dos recursos que serão necessários e assim também podendo esclarecer quaisquer dúvidas que possam surgir pela equipe de desenvolvimento. O *Product Owner* é o incumbido por elaborar o *Release Plan* e o *Product Backlog*, por ter conhecimento das prioridades que o empreendimento possui, cabe a ele atualizar constantemente o *Product Backlog (Planning)* para garantir que as funcionalidades mais relevantes sejam produzidas antes.

Scrum Master: Segundo Fonseca (2009) apud Machado e Medina (2009) é o responsável por assegurar que as boas práticas e regras do *Scrum* sejam utilizadas durante a realização do projeto, também garante que a equipe não se compactua em excesso referente a suas obrigações dentro de uma *Sprint*, caso seja necessário o *Scrum Master* também poderá realizar a remoção de empecilhos que sejam identificados pelo seu time durante as reuniões. É ele quem garante que o grupo esteja totalmente funcional e produtivo. É o guardião do processo, assegurando que as práticas do *Scrum* sejam utilizadas com a disciplina necessária e garante que o projeto e a cultura organizacional estejam ajustados para que as metas desejadas sejam alcançadas a cada *Sprint*.

Conforme Carvalho e Mello (2012) apud Lovato e Silva (2016) o *Scrum Master* possui autoridade direta sobre o processo *Scrum*, mas não sobre os membros das equipes. Ele deve treinar a equipe de desenvolvimento em termos de autogestão e que os integrantes possam tomar suas próprias decisões de ordem técnica na direção dos objetivos para a *Sprint*. Um *Scrum Master* é o porta voz da organização com relação ao *Scrum*, normatizando a comunicação para que as partes interessadas possam compreender o que vem sendo realizado. Este papel pode ser exercido por qualquer membro da equipe, mas, deve possuir características de comunicador e líder.

Scrum Team: De acordo com Fadel e Silveira (2010) é a composição na qual se organiza para poder alcançar o propósito que foi preestabelecido para ser entregue em uma *Sprint*. O time de desenvolvimento possui poder e controle para designar a maneira na qual o produto será projetado, também é de seu trabalho programar e avaliar o tempo necessário para a realização das tarefas a serem efetuadas durante a *Sprint*.

3.2 *Sprint*

Segundo Schwaber e Sutherland (2013) o *Sprint* é o principal evento do *Scrum*, é o período em que incrementos “prontos” são criados e que normalmente possuem durações entre um mês. Uma nova *Sprint* se inicia após a conclusão da *Sprint* anterior.

Conforme Schwaber e Sutherland (2013) cada *Sprint* possui uma definição do que deve ser construído. Durante o processo o escopo é detalhado e revisado entre o time e o *Product Owner*, podendo ser clarificado e renegociado, desde que não afete os objetivos inicialmente descritos. Além disso, não é recomendado alterar a composição do time ou a sua duração.

De acordo com Schwaber e Sutherland (2013) algumas situações podem gerar o cancelamento da *Sprint*, como por exemplo, quando os objetivos iniciais se tornam obsoletos. Porém, em função de sua curta duração é pouco provável que isto aconteça, e caso ocorra, esta tarefa cabe apenas ao *Product Owner*, mesmo que por recomendação das partes interessadas, do time de desenvolvimento ou do *Scrum Master*. Após o cancelamento, os itens do *Backlog* da *Sprint* que já haviam sido desenvolvidas podem ser entregues caso passe pela aprovação do *Product Owner*.

3.3 Valores e Pilares do *Scrum*

Segundo Schwaber e Sutherland (2013) o *Scrum* se apoia em 3 pilares para a sua implementação:

Transparência: devem ser visíveis todos os aspectos significativos do processo a todos os interessados. Assim como todos devem ter padrões comuns definidos. Um exemplo citado é a definição de “Pronto” que deve ser clara e comum a todos os envolvidos

Inspeção: os artefatos do *Scrum* e seus progressos devem ser inspecionados por seus usuários a fim de detectar variações. Não chegando ao ponto de ser excessiva causando prejuízo à execução das tarefas.

Adaptação: uma vez identificados desvios no progresso ou artefatos do *Scrum*, devem ser tomadas medidas para adaptação o mais rápido possível. A inspeção e adaptação podem ser realizadas formalmente nos eventos do *Scrum*.

3.4 Aplicação da Metodologia *Scrum*

Conforme Oliveira (2014) no decorrer do processo de evolução do modelo da Metodologia *Scrum* foram definidas uma série de premissas a serem criadas. Essas exigências são nomeadas como *Product Backlog*.

O *Product Backlog* possui uma lista de elementos que devem ser enfatizados pelo *Product Owner*, na qual está contido tudo que necessita ser efetuado, que possa ser relacionado com o valor de negócio da empresa, para a conclusão do projeto, englobando atributos funcionais ou não.

De acordo com Schwaber e Sutherland (2011) apud Oliveira (2014) o gerenciamento do *Backlog* do produto inclui: Relatar nitidamente os elementos do *Backlog* do Produto; Organizar os elementos do *Backlog* do Produto para atingir os objetivos; Afirmar o valor do serviço realizado pela equipe de desenvolvimento; Assegurar que o *Backlog* do produto seja perceptível, compreensível e claro para todos; Certificar-se de que a equipe de desenvolvimento entenda os elementos do *Backlog* do Produto no grau necessário.

Após isso surge o *Sprint Backlog*, que descreve o que terá que ser inserido em prática durante o projeto.

Segundo Nascimento (2012) apud Oliveira (2014) a maior parte dos itens pertencentes ao *Product Backlog* serão executados, mas para serem considerados parte do *Sprint Backlog* eles devem estar aptos, estimados e priorizados.

No *Scrum*, o avanço do projeto se fundamenta em interações bem definidas, nomeadas de *Sprint*, para cada *Sprint* aconselha a duração de 2 a 4 semanas período denominado como *Time Box*, as *Sprints* são compostas por quatro principais reuniões.

Incorporado às *Sprints* também existe o *Release Burndown Chart* que deve ser utilizado para o acompanhamento das atividades realizadas. No decorrer da fase de planejamento das *Sprints*, os integrantes da equipe *Scrum* determinam um certo valor para cada item e à medida em que é finalizado uma *Sprint*, os membros atribuem uma nova quantia de pontos, informando a quantidade que realmente foi necessária.

Essa forma de atualização do valor atribuído à atividade procura estabelecer uma visão analítica da progressão do projeto, além de permitir uma comparação entre o que foi planejado e o que foi realizado. Esse acompanhamento é muito importante para o desenvolvimento da

empresa com relação ao planejamento dos projetos seguintes, bem como pode ser utilizado para advertir o quanto a equipe está atrasada em cada iteração.

3.5 Reuniões

O *Scrum* é composto por quatro tipos de reunião, com a finalidade de organizar os processos, são elas:

- a) **Reunião de planejamento:** Segundo Schwaber e Sutherland (2013) é a reunião em que serão definidos os itens a serem desenvolvidos durante a *Sprint*, de forma colaborativa por toda a equipe de desenvolvimento. A mesma é dividida em duas etapas e tem uma duração média de oito horas para um *Sprint* de um mês, a qual deve ser proporcional ao tempo da *Sprint*.
- b) **Reunião diária:** Conforme Schwaber e Sutherland (2013) é um evento com 15 minutos de duração, com objetivo de que a equipe de desenvolvimento possa analisar o progresso do dia anterior. Resultando no planejamento do trabalho a ser realizado nas próximas 24 horas, devendo ser realizada sempre no mesmo local todos os dias, reduzindo a complexidade. Durante a reunião, cada membro do time de desenvolvimento deve responder às seguintes perguntas:
 1. O que foi feito desde a última reunião?
 2. O que será feito até a próxima reunião?
 3. O que está impedindo na conclusão do trabalho?

Com estas respostas avalia-se o progresso em direção ao objetivo final. Pode-se também identificar e remover qualquer problema que possa impactar na conclusão das atividades do backlog ao final do *Sprint*.

- c) **Reunião de revisão:** De acordo com Schwaber e Sutherland (2013) ocorre somente no fim do mesmo. Conta com a participação de toda a equipe do projeto e das partes interessadas, visando inspecionar o incremento gerado e adaptar o *Backlog* do produto, se for necessário.
- d) **Reunião de retrospectiva:** Conforme Schwaber e Sutherland (2013) do *Sprint* tem foco no processo, sendo uma oportunidade para toda a equipe *Scrum* se inspecionar, visando à criação de um plano de melhorias para os próximos

trabalhos, e dando ênfase na análise das pessoas, das relações, dos processos e das ferramentas.

4 RELATO TEÓRICO SOBRE IMPLANTAÇÃO

Durante o período de levantamento do material bibliográfico para este estudo foram identificados vários trabalhos que apontavam para situações de implantação do *Scrum*. Dentre eles, foi escolhido o descrito por Araújo e Dielle (2015) por ter uma abordagem que descrevia tanto benefícios quanto dificuldades, oferecendo assim uma visão um pouco mais ampla dos dois lados de um processo desse tipo. Outros trabalhos de implantação do *Scrum* podem ser encontrados em Conceição e Silveira (2015); Oliveira (2014).

Segundo o estudo de Araújo e Dielle (2015) a demanda por uma intervenção na forma como as atividades vinham sendo realizadas se deu por conta de problemas de retrabalho e menor entrega do que a necessária para os clientes de um ERP (*Enterprise Resource Planning*).

O processo se iniciou com um levantamento realizado por um consultor técnico, analisando as práticas de desenvolvimento da empresa e suas circunstâncias. Isso possibilitou reconhecer alguns problemas que ocasionaram a ineficácia na produção do *software* como, por exemplo, não obter controle eficiente nas alterações sobre o *software*, execução de múltiplas tarefas sem planejamento, alternância de funcionalidades de cargos, ausência de reuso de sub rotinas entre as equipes de desenvolvimento, sistema para controle das solicitações de mudanças encontrava-se em desenvolvimento, duplicidade em solicitações de funcionalidades e ajustes, inadequação nos padrões do projeto e falhas em tentativas de alteração do sistema legado.

Após o levantamento foi proposto pelo consultor a utilização da metodologia ágil *Scrum* na tentativa de suprir e resolver os impasses que foram identificados. A equipe compreendeu que o *Scrum*, sem uma certa adaptação não seria a solução perfeita para nenhuma empresa e por isso seria necessário adaptá-lo para sua prática, o que resultou em uma aceitação mais compreensível do que normalmente ocorre.

O processo se iniciou com a implementação de alguns métodos do *Scrum* na produção que já existia na empresa, uma vez que alterações teriam que passar pelo processo para que evitasse a entrega de algo inacabado ou que não se encontrava no padrão da empresa. Um dos métodos inseridos foi a definição da função de cada colaborador, também foi estabelecido um

tempo para que os mesmos pudessem realizar a troca de experiências definindo prioridades na execução das solicitações. A primeira *Sprint* teve o objetivo de liberar as requisições de mudanças que constavam em atraso.

Mesmo com o entusiasmo mostrado pela empresa às adequações propostas pela metodologia, complicações surgiram. A equipe definiu que as *Sprints* teriam a durabilidade de 15 dias e foi iniciada com 158 solicitações de mudanças, 11 colaboradores na equipe de desenvolvimento, gerente de setor de desenvolvimento conciliando papéis de *Scrum Master* e desenvolvedor e dono do produto com diversos papéis dentro da empresa, além de cerca de 314 HH (nomenclatura que definia o tempo estimado para a conclusão de uma solicitação de mudança, significando homem-hora).

Devido a quantidade enorme de HH definido na *Sprint*, existiam muitas solicitações alocadas na *Sprint* do que recursos humanos para as determinar. A primeira *Sprint* foi exclusivamente para o que necessitava ser realizado com maior urgência e não um prazo em que as tarefas que seriam executadas estavam planejadas.

Isto aponta para o fato de que a compreensão da equipe sobre o *Scrum* foi equivocada, uma vez que não ficou nítido que a *Sprint* precisava ser constituída por solicitações suscetíveis de serem concluídas e que precisavam ser primeiramente mensuráveis para determinar um prazo imposto através da equipe para ser efetuado. Os fatores que contribuíram para isso foi a grande quantidade de correções urgentes e divisão de tempo do *Scrum Master* entre suas duas funções (programador e gestor *Scrum*). Como resultado a *Sprint* acabou não sendo planejada da maneira devida. Era preciso desassociar a tarefa de *Scrum Master* da tarefa de desenvolvedor e designar a responsabilidade sobre cada uma delas a integrantes distintos.

Apesar do cálculo impreciso, a *Sprint* chegou perto de ser concluída, porém não é justificável o erro, apenas uma das solicitações comprometeu cerca de 28% do HH estipulado para a *Sprint* inteira, dessa maneira não estava sendo trabalhado a metodologia *Scrum*, pois o trabalho deveria estar sendo realizado de forma incremental, dividindo as tarefas em fragmentos para que exista supervisão gerando maior eficiência.

Um fator importante é que mesmo com a *Sprint* chegando próxima de ser concluída, o a quantidade de entregas não se aproximou da quantidade aprovações pelo departamento de qualidade, significando uma reprovação ampla, ou seja, que o time de desenvolvimento precisou executar várias vezes uma mesma solicitação até que ela fosse concluída com sucesso, levando mais tempo para serem entregues do que o estipulado. Para resolver, um novo

colaborador foi contratado para a equipe de desenvolvimento e a criados novos *status* para controlar de maneira mais eficaz as etapas dos processos, melhorando a informação sobre as etapas realizadas.

Devido ao problema de divisões de cargo que o *Scrum Master* sofria, foi preciso designar um colaborador apenas para o cargo específico. Com o esforço do novo *Scrum Master* aliado à decisão de incluir na *Sprint* apenas solicitações previamente discutidas com a equipe, foi observado que o número de solicitações de mudanças diminuiu.

A empresa realizou a troca de seu sistema legado, implementando uma nova maneira de desenvolvimento, tornando possível a manipulação de códigos de maneira simultânea, ou seja, mais de um desenvolvedor poderia manipular o código do mesmo projeto ao mesmo tempo, removendo uma limitação que existia. Essa mudança permitiu que a quantidade de solicitações aprovadas superasse a de concluídas indicando que algumas correções realizadas, utilizando esse novo método, acabaram ajustando mais de uma tarefa para uma única alteração efetuada.

5 RESULTADOS E DISCUSSÃO

A partir das descrições dos acontecimentos apresentados no estudo de Araújo e Dielle (2015) foi possível elaborar algumas considerações no sentido de ampliar as discussões e percepção de resultados.

Embora seja apenas um estudo, o que não o torna representativo e exige um maior levantamento de dados e aprofundamento para que se possa ter afirmações mais abrangentes, foi observado que a quantidade de solicitações corretivas, adaptativas ou evolutivas e o total de solicitações abertas no geral não teve uma distinção ampla.

O relato de Araújo e Dielle (2015) indicou diversos erros no uso da metodologia *Scrum*. Essas falhas atrapalharam a conclusão com sucesso das *Sprints* e por conta disso percebe-se que é necessário o comprometimento de todos os integrantes com os princípios da metodologia ágil. Além disso é preciso planejar monitorar, mensurar e adaptar de maneira correta, para se obter os ganhos esperados para os processos e para o desenvolvimento.

Quanto aos desenvolvedores que cometiam falhas em seus serviços, uma maneira de impedir esse obstáculo é os próprios realizem uma inspeção detalhada antes de ser realizado o envio das alterações para a equipe de testes, fazendo assim uma revisão. Desse modo seria possível a identificação de alguns pequenos erros previamente, porém, a equipe de testes é

imprescindível pois gera uma excelência no desenvolvimento e padrões, evitando que problemas maiores aconteçam no futuro.

O estudo de Araújo e Dielle (2015) também indicou que a designação de papéis e melhoria do sistema de controle foram fundamentais para estabelecer o cenário positivo para a empresa, embora essas mudanças pudessem ter ocorrido de forma menos tardia se a empresa já utilizasse as metodologias ágeis anteriormente, o que reduziria falhas e possíveis mudanças em um período menor.

Para amenizar os problemas na aplicação do *Scrum*, é essencial que seja mensurado em alerta o que está se passando na empresa durante sua execução. Não mensurar o *Scrum* acaba gerando dúvidas sobre sua eficácia. Alguns métodos simples como obter informações de datas de chamadas registradas pelo suporte referente a problemas podem servir de indicadores para contestar informações ocorridas nas *Sprints* anteriores e a identificar falhas que não haviam sido notadas, gerando um ciclo favorável para o próprio *Scrum*.

A falta de planejamento inicial e do monitoramento foi, no estudo relatado por Araújo e Dielle (2015), determinante para um resultado inicial abaixo do esperado pela empresa.

6 CONSIDERAÇÕES FINAIS

A inserção da metodologia *Scrum* pode parecer simples, porém demanda uma boa compreensão por parte da equipe, podendo exigir a mudança de hábitos que eram consistentes dentro da empresa.

Apesar dos inúmeros benefícios que o *Scrum* pode trazer, nota-se que quando esta metodologia não é seguida corretamente pode não se obter o sucesso previsto em sua implementação. Por outro lado, o *Scrum* apresenta metodologia flexível que possibilita adaptação até mesmo em sua utilização de forma conjunta com outras ferramentas de gerenciamento de projetos, demonstrando ser muito eficaz se empregada ao menos seus fundamentos de maneira correta seguindo os princípios básicos do *framework*.

Conclui-se assim que mesmo nos casos em que não se consiga o resultado esperado em sua plenitude, o *Scrum* é um excelente método para alinhar a equipe e tornar o ambiente de desenvolvimento mais democrático, tentando compor um ambiente colaborativo por meio do qual se torna mais fácil localizar falhas e conseqüentemente uma solução mais rápida e eficiente.

REFERÊNCIAS

- ARAÚJO, M. A.; DIELLE, D. F. P. **Desafios na Implementação do Scrum: um Estudo de Caso Sobre a Utilização da Metodologia Ágil em uma Empresa Desenvolvedora de Software**. Publicado em 2015. Disponível em: <<https://seer.cesjf.br/index.php/cesi/article/view/516/401>>. Acesso em: 10 de out. de 2020.
- CONCEIÇÃO, J.; SILVEIRA, S. R. **Aplicação de Metodologias Ágeis para Desenvolvimento de Software: um Estudo de Caso na Empresa Alliance Software**. UNIVERSIDADE FEDERAL DE SANTA MARIA. Publicado em 09 de dez de 2015. Disponível em: <https://repositorio.ufsm.br/bitstream/handle/1/12846/TCCG_SIFW_2015_CONCEICAO_JULIANO.pdf?sequence=1&isAllowed=y>. Acesso em: 10 de out. de 2020.
- EXTERKOETTER, F. **Blendwork: Framework Orientado a Objetos para Desenvolvimento Rápido de Aplicações Comerciais Cliente/Servidor**. Publicado em 2003. Disponível em: <<https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/86053/194492.pdf?sequence=1&isAllowed=y>>. Acesso em: 24 de maio. de 2021.
- FADEL, A. C.; SILVEIRA, H. M. **Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum e Lean**. Universidade Estadual de Campinas. Publicado em 2010. Disponível em: <https://www.academia.edu/617187/Metodologias_%C3%A1geis_no_contexto_de_desenvolvimento_de_software_XP_Scrum_e_Lean>. Acesso em: 10 de out. de 2020.
- LOVATO, A. L.; SILVA, E. C. **FRAMEWORK SCRUM: EFICIÊNCIA EM PROJETOS DE SOFTWARE**. Revista de Gestão e Projetos - GeP. Publicado em 15 de mar de 2016. Disponível em: <<https://periodicos.uninove.br/gep/article/view/9640/4385>>. Acesso em: 10 de out. de 2020.
- MACHADO, M.; MEDINA, S. G. **SCRUM – Método Ágil: uma mudança cultural na Gestão de Projetos de Desenvolvimento de Software**. REVISTA Intraciência Uniesp. Publicado em 2009. Disponível em: <http://uniesp.edu.br/sites/_biblioteca/revistas/20170531154126.pdf>. Acesso em: 10 de out. de 2020.
- MARTINS, L.; ROCHA, M.; SANTOS, M.; SAVOINE, M. **Análise de Gerenciamento de Projeto de Software Utilizando Metodologia Ágil XP e Scrum: Um Estudo de Caso Prático**. Centro Universitário Luterano de Palmas. Publicado em 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: 10 de out. de 2020.
- OLIVEIRA, J. F. **A utilização da metodologia Scrum sob a percepção da equipe de desenvolvimento em uma empresa privada de software: Um estudo de caso**. Universidade

Federal da Paraíba. Publicado em 2014. Disponível em: <<https://si.dcx.ufpb.br/wp-content/uploads/2015/12/Jessyca-Ferreira-de-Oliveira.pdf>>. Acesso em: 10 de out. de 2020.

RACHID, C. L.; STOPA, G. R. **SCRUM: METODOLOGIA ÁGIL COMO FERRAMENTA DE GERENCIAMENTO DE PROJETOS**. CES REVISTA. Publicado em 2019. Disponível em: <<https://seer.cesjf.br/index.php/cesRevista/article/view/2026/1315>>. Acesso em: 10 de out. de 2020.

SCHWABER, J.; SUTHERLAND, K. **Um guia definitivo para o Scrum: As regras do jogo**. CES REVISTA. Publicado em jul de 2013. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 10 de out. de 2020.