

JOSSAN SOBRINHO PETRENKO

**Metodologia SCRUM**:

Metodologia ágil para gestão e planejamento de projetos de software

Cuiabá

2018

jossan sobrinho petrenko

**Metodologia SCRUM**:

Metodologia ágil para gestão e planejamento de projetos de software

Projeto apresentado ao Curso de Sistema de Informação da Instituição Unic.

Orientador: Felipe Douglas Machado da Cunha

Cuiabá

2018

JOSSAN SOBRINHO PETRENKO

**METODOLOGIA SCRUM:**

Metodologia ágil para gestão e planejamento de projetos de software

Trabalho de Conclusão de Curso apresentado à UNIC, como requisito parcial para a obtenção do título de graduado em Sistema de Informação.

**BANCA EXAMINADORA**

Prof(a). Titulação Nome do Professor(a)

Prof(a). Titulação Nome do Professor(a)

Prof(a). Titulação Nome do Professor(a)

Cuiabá, dia de mês de 2018

Dedico este trabalho aos meus pais.

PETRENKO, Jossan Sobrinho. Metodologia SCRUM: Metodologia ágil para gestão e planejamento de projetos de software. 2018. Número total de folhas. Trabalho de Conclusão de Curso (Graduação em Sistema de Informação) – UNIC, Cuiabá, 2018

**RESUMO**

Com as empresas demandando formas mais ágeis, eficientes e que garantam uma entrega de qualidade que atenda as reais necessidades dos seus clientes em tempo hábil, ao gerenciar seu(s) times(s), este trabalho tem por objetivo explicitar o que é e como funciona a metodologia de desenvolvimento ágil SCRUM, dessa forma é apresentando suas principais características, funções, valores, os artefatos que ela utiliza e gera, seus eventos durante cada ciclo de iteração, os papeies que são necessários na utilização dessa metodologia, e como se dá sua implantação e funcionamento. Para tal, foram efetuadas pesquisas realizadas em livros e artigos de cunho acadêmico e educacional voltados para ao tema em questão e da maneira que uma metodologia ágil deve ser estruturada para ser considerada como tal, e que apresentassem as principais vantagens e desvantagens de sua aplicação dentro de um ambiente corporativo. Dessa maneira gerando um entendimento conciso sobre a metodologia ágil para gestão e planejamento de projetos de software SCRUM.

**Palavras-chave:** Ágil; Desenvolvimento; Metodologia; SCRUM; Vantagens.

PETRENKO, Jossan Sobrinho. SCRUM Methodology: Agile methodology for management and planning of software projects. 2018. Total number of sheets. Course Completion Work (Graduation in Information System) – UNIC, Cuiabá, 2018

**ABSTRACT**

With companies demanding more agile, efficient and guaranteeing a quality delivery that meets the real needs of their customers in a timely manner, when managing their team (s), this work aims to make explicit what is and how the SCRUM agile development methodology, in this way it is presenting its main characteristics, functions, values, the artifacts it uses and generates, its events during each iteration cycle, the roles that are necessary in the use of this methodology, and how it happens its implementation and operation. For that, research was carried out on books and articles of an academic and educational nature focused on the subject in question and how an agile methodology should be structured to be considered as such and present the main advantages and disadvantages of its application within of a corporate environment. In this way generating a concise understanding of the agile methodology for SCRUM software project management and planning.

**Keywords**: Advantages; Agile; Development; Methodology; SCRUM.

**LISTA DE ILUSTRAÇÕES**

**Figura 1** – Burndown Chart 21

**Figura 2** – Processos de uma Sprint 23

**LISTA DE TABELAS**

**Tabela 1** – Manifesto de Desenvolvimento Ágil para Software 16

**LISTA DE ABREVIATURAS E SIGLAS**

*PoC Proof of concept* (Prova de conceito)

**SUMÁRIO**

[1 INTRODUÇÃO 12](#_Toc524634265)

[2 METODOLOGIA SCRUM: APRESENTAÇÃO 15](#_Toc524634266)

[2.1 declaração de princípios - *AGILE MANIFESTO* 16](#_Toc524634267)

[2.2 O Controle Empírico de Processo 18](#_Toc524634268)

[2.2.1 Pilar transparência 18](#_Toc524634269)

[2.2.2 Pilar inspeção 18](#_Toc524634270)

[2.3 ARTEFATOS 19](#_Toc524634271)

[2.3.1 O *product backlog* 19](#_Toc524634272)

[2.3.2 O *sprint backlog* 19](#_Toc524634273)

[2.3.3 O *burndown* 20](#_Toc524634274)

[2.4 EVENTOS 21](#_Toc524634275)

[2.4.1 A definição da *sprint* 21](#_Toc524634276)

[2.4.2 A *daily* scrum 23](#_Toc524634277)

[2.4.3 A *sprint review* 24](#_Toc524634278)

[REFERÊNCIAS 26](#_Toc524634279)

# 1 INTRODUÇÃO

Para Schwaber e Sutherland (2013), em gestão e planejamento para projetos de software, o levantamento de requisitos tende possuir alterações constantes, por conta da grande demanda e enorme pressão do mercado por melhorias e aumento de produtividade, e isso afeta diretamente o processo desenvolvimento de software, tornando esse processo desafiador, principalmente para pequenas empresas. Neste cenário, viu-se a necessidade de uma nova abordagem metodológica em gestão e planejamento para projetos de software, conhecidas como metodologias ágeis, ramo da Engenharia de Software, que tem como objetivo final aumentar o nível de assertividade, produtividade e qualidade, bem como a satisfação do cliente, no processo de desenvolvimento de software.

No início da década de 90, a comunidade já fomentava novas metodologias para gestão e planejamento para projetos de software, mas foi no ano de 1995 que nasceu a metodologia ágil conhecida como Scrum, e seus autores Jeff Sutherland e Ken Schwaber, usam do termo “Scrum” presente no artigo “*The New Product Development Game*” redigido por Takeuchi e Nonaka(1986). O artigo em questão cria um contraponto entre uma partida de *Rugby* e o processo de gestão e planejamento para um produto, através do uso de táticas de *Rugby*, como trabalho em equipe, e nele os autores concluem que pequenos grupos autogeridos possuem índices de desempenhos melhores ao criar produtos no momento em que apenas objetivos são elencados e não as tarefas. Dessa forma, o Scrum usa de ferramentas para abordar o planejamento e execução de maneira incremental e repetitiva, tendo como seu cerne ideológico os processos empíricos, fracionando objetivo em objetivos menores tendo como o foco a entrega de software que atenda as especificações. Ainda Schwaber e Sutherland (2013), a metodologia Scrum prima por uma aproximação e colaboração maior entre a equipe técnica de desenvolvimento e a equipe responsável pela regra negocial para que a execução do produto final atenda aos objetivos, dessa forma procure-se a redução de riscos associados as incertezas do projeto, facilitar as mudanças necessárias durante o processo gestão, planejamento e desenvolvimento do software com o objetivo final de satisfação do cliente ao receber um de alta qualidade e útil para seus fins.

Dentro das metodologias ágeis, a metodologia Scrum se destaca através de seus valores e princípios que o torna um framework adaptável no suporte de planejamento e gerenciamento de projetos de alta ou baixa complexidade, fazendo do Scrum um framework de extrema de relevância no mercado atual. Este trabalho tem como foco um entendimento mais profundo sobre o Scrum, suas ferramentas, como elas funcionam, seu ciclo iterativo, bem como suas vantagens e desvantagens.Com isso gerar um entendimento maior e mais profundo para a comunidade, sobre essa metodologia que é largamente utilizado pela comunidade de desenvolvimento de software, que utiliza a abordagem ágil de desenvolvimento, facilitando o entendimento sobre o Scrum em um único artigo, pois existe uma perceptível dificuldade em encontrar matérias que abordam esse conteúdo de maneira mais ampla e profunda.

A necessidade de adaptação e o alto desempenho requisitado por parte das empresas de desenvolvimento de software pelo mercado que necessitam de uma nova maneira de se organizar e ao mesmo tempo atender as necessidades de seus próprios cliente foram a motivação para esse trabalho, exibindo as características marcantes do Scrum e seu funcionamento em uma empresa para aqueles que tem interesse em conhecer sobre essa metodologia ágil de desenvolvimento de software. Como se caracteriza a metodologia Scrum? De que maneira esta metodologia de modelo ágil pode trazer benefícios para gestão, planejamento para projetos de software e desenvolvimento de software? De quais vantagens pode ser beneficiar ao usar a metodologia dentro de um projeto de desenvolvimento de software? Ao decorrer do artigo o Scrum e suas características principais, suas ferramentas, funcionamento e sua aplicação dentro de um projeto serão apresentados, além da mostra de como e feita de sua utilização em uma empresa de desenvolvimento de software.

O seguinte trabalho tem como objetivo analisar as características e valores que a metodologia ágil possuí, através de pesquisas de literatura em diferentes publicações sobre o tema. Descrever o funcionamento de cada passo presente dentro da metodologia Scrum, quando inserida em um projeto. Enfatizar a importância da metodologia Scrum no processo de gestão, planejamento e desenvolvimento de software em uma empresa. Apresentar as principais vantagens e desvantagens no uso metodologia Scrum no processo de gestão, planejamento e desenvolvimento de software em uma empresa.

Neste trabalho será usado técnicas de pesquisas para sua elaboração, com o uso de leitura de livros, artigos e trabalhos que envolva uma perspectiva acadêmica e educacional sobre a metodologia Scrum, datados a partir da década de 90 até então, e será usada a internet como fonte facilitadora para encontro de material. Estas pesquisas usam como base artigos sobre a metodologia de desenvolvimento ágil, com seu foco voltado para as características do Scrum, e em pesquisas nas quais essa metodologia foi verificada. Neste trabalho foi feito o uso de uma revisão narrativa de literatura, com a descrição e discussão sobre o assunto, sob um ponto de vista teórico e contextual, sua composição básica e feita com analisa de literaturas publicada em livros, artigos de revista eletrônica, na interpretação e analise critica pessoal do autor. Diferentes artigos foram levantados através das bibliotecas eletrônicas SciELO e Google Acadêmico.

.

# 2 METODOLOGIA SCRUM: APRESENTAÇÃO

A definição de SCRUM é: “Um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível.” (Guia do Scrum, 2013, p.03).

Em Schwaber e Beedle (2002), por ser coesa e com baixo acoplamento essa metodologia, tem condição de ser aplicada a qualquer produto que seja desenvolvido que tenha a necessidade de um grande volume de trabalho intelectual. Se enquadra na categoria de metodologia ágil, devido ao seu modelo possuir a características “adaptação” e “flexibilidade”, seja devido às necessidades do cliente, do produto, ou das situações que podem surgir durante o planejamento e desenvolvimento de um software.

Para Schwaber e Beedle (2002), pode-se ter a seguinte perspectiva perante o Scrum, que ele é um framework que nos mostra “o que se fazer”, e não “como fazer”. Tem como foco integrar e tornar todo o time responsável pelo desenvolvimento do produto e as necessidades do cliente, ao invés de se guiar pelo modelo tradicional de “passo-a-passo” das abordagens sequenciais, como modelo cascata, que tem uma perspectiva mais inflexível no processo de planejamento e desenvolvimento do produto.

Sutherland e Schwaber (2013), diz que o Scrum não é uma técnica para a construção de produtos nem um processo engessado, e sim uma estrutura maleável, permitindo a inserção de técnicas e processos, fazendo dela uma metodologia adaptável a realidade da empresa, produto ou cliente. Dessa maneira, o Scrum tornou-se eficaz em suas práticas para desenvolvimento.

Para Sutherland e Schwaber (2013), Scrum possui uma estrutura simples e adaptável. Em uma explicação breve é possível para entender seu funcionamento. Porém, existe a questão de “como implementá-lo” de maneira correta em um projeto. Pode-se fazer a seguinte analogia: Os valores do Scrum são como a fundação e estrutura de um prédio. Não pode ser alterado ou ignorado, se o fizer, irá impactar diretamente na estrutura do Scrum, podendo causar um colapso. Entretanto, há a possibilidade que o interior e fachada seja moldado de acordo com as necessidades do produto em questão, inserindo recursos e artefatos que agreguem valor ao desenvolvimento do produto.

# 2.1 declaração de princípios - *AGILE MANIFESTO*

O Scrum é definido como “metodologia ágil de desenvolvimento de software”, e o conhecimento de seus pilares se faz necessário. O Scrum, se guia através das metodologias do Agile Manifesto publicado em 2001 por 17 desenvolvedores. Neste manifesto estas suas principais ideias, e elas são:

**Tabela 1:** Manifesto de Desenvolvimento Ágil para Software

|  |  |
| --- | --- |
| **Valorizamos** | **Mais que** |
| Indivíduos e interações | processos e ferramentas |
| Software em funcionamento | documentação abrangente |
| Colaboração com o cliente | negociação 7de contratos |
| Responder a mudanças | seguir um plano |

**Fonte**: BECK et al., 2001

O Agile Manifesto possui 12 princípios, sendo eles (BECK et al., 2001):

1. A prioridade é satisfazer o cliente por meio da entrega rápida e contínua, agregando valor ao software.
2. Alterações nos requisitos são bem recebidas, ainda que sejam feitas tardiamente no desenvolvimento. Processos ágeis tiram vantagem destas alterações objetivando vantagem competitiva para o cliente.
3. Entregar o software funcionando para o cliente com maior frequência, minimizando o prazo de entrega, prezando poucas semanas ou poucos meses, sempre buscando o menor prazo.
4. Patrocinadores, clientes e desenvolvedores devem trabalhar diariamente em parceria durante o andamento do projeto.
5. Construir projetos com a equipe motivada, prover o ambiente e o suporte necessário para o desenvolvimento do trabalho, bem como confiar na equipe para a realização do mesmo.
6. O método mais eficiente e eficaz de conduzir informações para e entre uma equipe de desenvolvimento é através de conversas pessoalmente.
7. Software operando o é a medida mais importante de progresso
8. Processos ágeis favorecem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e clientes devem ser habilitados a manter um ritmo constante indefinidamente.
9. Atenção constante à excelência técnica e design adequado aumentam a agilidade.
10. Simplicidade -- a arte de potencializar a quantidade de retrabalho não efetuado -- é essencial.
11. As melhores arquiteturas, requisitos e designs surgem de equipes auto organizáveis.
12. Em intervalos frequentes, a equipe reflete sobre como se tornar mais eficaz e então aprimora e ajusta seu comportamento conforme a necessidade.

Segundo Schwaber e Beedle (2002), com o uso desses 12 princípios as metodologias chamadas ágeis ou agilistas são geradas, e com o uso desses mesmos princípios sua aprimoração se dá de uma forma continuada pelos próprios times que as implementam, o que tem tornado os debates, workshop e oficinas sobre esse tema algo com bastante adesão dentro dessas comunidades chamadas agilistas, uma vez que essas metodologias tendem por si próprias a serem revistas e aperfeiçoadas.

# O Controle Empírico de Processo

O *SCRUMstudy*¹ informa que, no Scrum, usa-se a ideia de que o conhecimento é acumulado de maneira empírica, ou seja, se dá através de experimentação e observação e não com um planejamento detalhado, feito de modo antecipado. O Controle Empírico de Processo, tem como sua base 3 pilares: Transparência, inspeção e adaptação.

# 2.2.1 Pilar transparência

Conforme Schwarz e Sutherland (2013) a transparência é o que gera a habilidade que todo e qualquer aspecto do processo seja observado e conhecido por qualquer indivíduo que esteja incluído ao processo de desenvolvimento do produto em questão.

A transparência deve estar em toda confecção do *Backlog* de atividades criado, e que este esteja disponível para qualquer envolvido no processo, ou em reuniões que buscam revisar tudo que já foi feito, para que assim todos os envolvidos estejam cientes do andamento do projeto.

# 2.2.2 Pilar inspeção

De acordo Schwaber e Sutherland (2013), o produto deve estar sob constante análise dos stakeholders e usuários, e são conhecidos como intervenientes, dessa forma todo o feedback necessário sobre tudo o que foi feito até o dado momento, e as informações do que ainda estará por vir serão analisadas e controladas.

¹ Disponível em:

http://www.scrumstudy.com/blog/why-is-empirical-process-control-so-important-in-scrum- projects/

quando o Scrum Team e os stakeholders conhecem através dos itens anteriormente mencionados, transparência e inspeção, seus processo, ferramentas e artefatos utilizados no processo de desenvolvimento do produto, e dessa forma adaptam a forma como trabalham e fazendo ajustes com a intenção de melhorar a cadeia de trabalho sempre que for observado necessário.

# 2.3 ARTEFATOS

# 2.3.1 O *product backlog*

Em “*The Scrum Guide*”(2013), para seus autores, “O *Product Backlog* é uma lista de tudo o que pode ser necessário no produto e é a única fonte de requisitos para quaisquer alterações a serem feitas ao produto.”

Em Sutherland e Schwaber (2013), o *Backlog* do produto é um dos primeiros passos a ser efetuados para a implementação do Scrum em um projeto.

Ainda Sutherland e Schwaber (2013), diz que o *Backlog* é gerado através do *Product Owner*, lá estará listada todas atividades e requisitos que devem ser concretizadas pelo time durante o projeto, sejam requisitos funcionais ou não funcionais, são organizadas através de prioridades definidas pelo próprio *Product Owner* no momento de levantamento de requisitos. Uma vez confeccionado, está sempre passível de alterações quando essas são julgadas necessárias, de maneira a adicionar, remover ou alterar atividades no *Backlog.*

# 2.3.2 O *sprint backlog*

*Sprint Backlog* segundo Sutherland e Schwaber (2013), é uma lista ordenada de itens, itens que representam atividades que devem ser realizadas durante o período da *Sprint*, gerada geralmente a partir de uma parte do *Product Backlog,* também podendo ser todo o backlog, quando o backlog é muito pequeno, como em projeto do tipo *PoC*, ou mesmo o que resta do Backlog em uma fase final do projeto.

No *Sprint Backlog* para Sutherland e Schwaber (2013), estão atividades que o time mensura ter a capacidade se concluir durante a próxima Sprint. Os itens são movidos do *Product Backlog* para o *Sprint Backlog* de acordo com a prioridade estabelecida pelo *Product Owner* e a percepção do Time Scrumdo tempo necessário para realização das funcionalidades, no momento da reunião de planejamento da Sprint. Esse é um artefato, que analisado ao decorrer das *Sprints*, faz o cliente entender de maneira bem objetiva a quantidade que o time que ele tem é capaz de produzir.

# 2.3.3 O *burndown*

O *Burndown* se trata de acordo com Sutherland e Schwaber (2013), de uma representação gráfica onde aponta a execução corrente de uma *sprint*, tendo uma relação entre o que é necessário ser feito e o tempo restante disponível para sua realização. Possui no eixo X, o eixo horizontal, o total de dias da Sprint corrente, e no eixo Y, o eixo vertical, a quantidade de trabalho a ser realizado.

Ainda Sutherland e Schwaber(2013), diz que através do *Burndown* gerasse uma média ideal de execução de trabalho em um determinado tempo, posteriormente realizasse uma comparação com o desempenho de todo o time, permitindo a visualização de como o time progride durante o andamento do projeto.

**Figura 01:** Burndown Chart



**Fonte:** [Hans-Petter Halvorsen](http://home.hit.no/~hansha/hph.htm), (2013) ²

Para Sutherland e Schwaber(2013), o gráfico de *Burndown* permite um fácil o acompanhamento da produtividade do time e durante a iteração. Se existe um desvio acima da reta ideal, significa que a equipe está produzindo menos do que deveria, dessa maneira, pode não acabar as tarefas iteração corrente. De outra forma, se o desvio ocorre abaixo da reta ideal, a equipe está produzindo melhor do que o previsto, o que pode relatar uma subestimação da produtividade do time ou uma superestimação da complexidade das tarefas da iteração.

# 2.4 EVENTOS

# 2.4.1 A definição da *sprint*

Schwaber e Sutherland (2013), diz que o “coração” do Scrum é a Sprint. Em um intervalo próximo a um mês, é executado um incremento ao produto “pronto”,

² Disponível em:

https://www.halvorsen.blog/documents/programming/software\_engineering/resources/resources/Scrum%20Overview.pdf

uma versão proveitosa, de preferência que já agregue valor ao usuário final, de produto que está em processo de concepção. As sprints através de todo esse processo criação, tem uma duração consistente no período de esforço de desenvolvimento. Uma nova sprint tem início logo após a finalização da sprint passada. Ao finalizar o *Product Backlog*, projeto está concluído, não sendo necessário uma nova *Sprint*.

Schwaber e Sutherland (2013), diz que primariamente, ao se começar uma Sprint, é indispensável que o *Product Owner* informar a todos o que há para ser executado no projeto, sem a obrigação de relatar todos os itens que produzem o *backlog*, focando nos itens de maior prioridade.

Em Schwaber e Sutherland (2013), após esse passo, todo o time deve reunir e estabelecer um objetivo particular para aquela Sprint, requisitos que sirvam de consulta caso haja dúvidas no futuro sobre o objetivo dessa *Sprint*. Neste momento toda a equipe deverá estar completamente comprometida com o objetivo da *Sprint*, pois é neste momento em que terão a informação do que é necessário ser executado, entender prioridades e dessa forma analisar o que são capazes de entregar dentro do prazo estabelecido da *Sprint*, podendo haver negociação de prazos ou atividades com o Product Owner caso haja necessidade, este é o momento de fazer *Product Owner* entender que a quantidade de recursos para execução do backlog normalmente é limitada.

Com prioridade e escopo estabelecido, Schwaber e Sutherland (2013), prossegue dizendo que time estará diariamente se informando sobre a execução da *Sprint* através das *Dailys* Scrums e ficarão cientes do volume ainda necessário de trabalho ser executado dentro do tempo disponível. Mesmo o Scrum sendo uma metodologia que adeque e encoraje mudanças de forma mais flexível, essas mudanças não devem ser feitas durante a *Sprint* corrente, e sim em uma próxima *Sprint*. Dessa forma, requer que durante a reunião de *Sprint* todos estejam conscientes do que precisa ser executado, das prioridades e a capacidade individual e geral do time.

**Figura 02:** Processos de uma Sprint



**Fonte:** Tomas Anderson Souza Silva, (2016) ³

Em concordância com os autores Sutherland e Schwaber (2013), as sprints devem ter tempo limitado, e no caso de sua duração torna-se muito longa, a definição do que está sendo executado pode ter mudado, podendo intensificar o grau de complexidade e o risco da execução. Ademais para os autores, as sprints favorecem a previsibilidade, enfretamento de riscos e limitam o risco de custo para um mês.

# 2.4.2 A *daily* scrum

Segundo Schwaber e Sutherland (2013), *Daily* Scrum devem ser reuniões realizadas diariamente com curta duração tempo, aproximadamente entre 10 a 15 minutos, com a intenção de apresentar tudo o que foi executado no dia anterior, informar problemas e impedimentos encontrados e definir qual será o foco e ações de trabalho do dia corrente.

³ Disponível em:

http://tassinfo.com.br/gestao-de-produto/scrum

Schwaber e Sutherland (2013), relata que problemas e/ou impedimentos relatados na *Daily* Scrum, não devem ser detalhados e debatidos durante essa reunião, e sim posteriormente. A intenção dessa reunião não é resolver essas situações, e sim para cada membro do time informar sua situação na atividade corrente. Se houver problemas, após a reunião diária, um grupo reduzido de membros do time, que detém o conhecimento preciso para sana-los ou que sejam os diretamente envolvidos com essa situação podem se reunir, dessa forma compartilhar informação e conhecimento, e assim achar uma melhor solução viável.

Em Schwaber e Sutherland (2013), para uma melhor realização da *Daily* Scrum, cada membro do time deve responder individualmente a 3 essas perguntas pré-definidas:

* “O que fiz ontem?”
* “O que farei hoje?
* “Há algo me impedindo de progredir?”

Dessa maneira Schwaber e Sutherland (2013), diz que o time conhecera o status de execução do projeto de uma maneira geral. Terá a informação de onde existem problemas e onde o trabalho está sendo executado de maneira esperada, possibilitando que o time tenha uma postura proativa e tome as medidas necessárias perante o problema, já que este torna-se um problema do time.

Schwaber e Sutherland (2013), deixa bem claro que a *Daily* Scrum não deve ser vista como um momento para relatar ao superior ou chefe quem está atrasado ou não, e sim informar em que status está andamento da atividade de cada membro do time dentro do projeto, e estipular compromissos entre o próprio time.

# 2.4.3 A *sprint review*

Em o Guia do Scrum(2013), a *Sprint Review* é realizada ao termino de uma *Sprint*, demonstrando tudo o que foi executado, bem como as atividades que foram feitas e se o objetivo pré-estabelecido na reunião de *Sprint* foi cumprido.

Para Schwaber e Sutherland (2013), neste ponto, o cenário ideal é que o time consiga executar todas as atividades que foram movidas do *Product Backlog* para o *Sprint Backlog*, mas o foco principal é verificar se objetivo geral definido para aquela sprint foi alcançado, e se não, quais ações devem ser realizadas para que isso não ocorra novamente.

# REFERÊNCIAS

BECK, K. et al. The agile manifesto. 2001. Disponível em: <http://www.agilemanifesto.org/>. Acesso em: 03 setembro 2018.

SCHWABER, Ken; SUTHERLAND, Jeff. **Guia do Scrum.** [S.l.].2013. Disponível em: <https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf Acesso em: 03 setembro 2018.

SCHWABER, Ken.; BEEDLE, Mike. Agile Software Development with Scrum, Upper Saddle River: Prentice Hall. 2002. p. 158 Disponível em: <http://dl.acm.org/citation.cfm?id=559553> Acesso em: 03 setembro 2018.

SUTHERLAND, Jeff. Agile Can Scale: Inventing and Reinventing SCRUM in Five Companies. Cutter it Journal. 2001, v. 14 n. 12 Disponível em: <https://www.researchgate.net/publication/290823579\_Agile\_Can\_Scale\_Inventing\_and\_Reinventing\_SCRUM\_in\_Five\_Companies> Acesso em: 03 setembro 2018.

TAKEUCHI; NONAKA. The New New Product Development Game. Harvard Business Review 64, no. 1, 1986.