



---

JOSSAN SOBRINHO PETRENKO

---

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, XX de dezembro 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 papéis 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

<b>Figura 1</b> – Burndown Chart.....	21
<b>Figura 2</b> – Processos de uma Sprint .....	23
<b>Figura 3</b> – Papéis do <i>Scrum Master</i> .....	28

## LISTA DE TABELAS

<b>Tabela 1</b> – Manifesto de Desenvolvimento Ágil para Software .....	16
---	----



## LISTA DE ABREVIATURAS E SIGLAS

DDS        Desenvolvimento Distribuído de Software

DoD        *Definition of Done* (Definição de pronto)

PoC        *Proof of concept* (Prova de conceito)

P.O.        *Product Owner* (Dono do produto)



## SUMÁRIO

<b>1 INTRODUÇÃO</b>	12
<b>2 METODOLOGIA SCRUM: APRESENTAÇÃO</b>	15
2.1 DECLARAÇÃO DE PRINCÍPIOS - <i>AGILE MANIFESTO</i>	16
2.2 O CONTROLE EMPÍRICO DE PROCESSO	18
2.2.1 O Pilar transparência	18
2.2.2 O Pilar inspeção	18
2.2.3 O Pilar adaptação	19
2.3 OS ARTEFATOS	19
2.3.1 O <i>product backlog</i>	19
2.3.2 O <i>sprint backlog</i>	20
2.3.4 O <i>burndown</i>	20
2.4 OS EVENTOS	21
2.4.1 A definição de <i>sprint</i>	21
2.4.2 A <i>daily Scrum</i>	23
2.4.3 A <i>sprint review</i>	24
<b>3 METODOLOGIA SCRUM: IMPLEMENTAÇÃO</b>	26
3.1 OS PAPÉIS DA METODOLOGIA SCRUM	26
3.1.1 O <i>product owner</i>	26

3.1.2 O <i>Scrum master</i>	27
3.1.3 O <i>Scrum team</i>	29
3.2 O MODELO CÍCLICO ITERATIVO DO SCRUM	30
3.3 A ABORDAGEM DISTRIBUÍDA PARA O SCRUM	31
3.3.1 O <i>Scrum of scrums</i>	32
<b>4 METODOLOGIA SCRUM: UMA PERSPECTIVA GLOBAL</b>	34
4.1 AS VANTAGENS INERENTES AO SCRUM	34
4.2 AS POSSÍVEIS DESVANTAGENS NO SCRUM	35
4.3 OS ASPECTOS DA TRANSPARÊNCIA NO SCRUM	37
4.3.1 O conceito “Pronto”, sua definição	38
4.3.2 A estimativas de atividades e suas dificuldades	39
4.3.2.1 O <i>planning poker</i> e seu funcionamento	40
<b>5 CONSIDERAÇÕES FINAIS</b>	43
<b>REFERÊNCIAS</b>	45

## 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 comercial para que a execução do produto final atenda aos objetivos, dessa forma procure-se a redução de riscos associados às 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 relevância no mercado. 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 têm 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 é 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 possui, 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 é feita com análise de literaturas publicada em livros, artigos de revista eletrônica, na interpretação e análise crítica 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**

<b>É valorizado</b>	<b>Mais que</b>
Indivíduos e interações	processos e ferramentas
Software em funcionamento	documentação abrangente
Colaboração com o cliente	negociação de 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 emergem 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 adesão dentro dessas comunidades chamadas agilistas, uma vez que essas metodologias tendem por si próprias a serem revistas e aperfeiçoadas.

## 2.2 O CONTROLE EMPÍRICO DE PROCESSO

O *SCRUMstudy*<sup>1</sup> 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 O 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 O 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 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

---

<sup>1</sup> Disponível em:

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

forma como trabalham e fazendo ajustes com a intenção de melhorar a cadeia de trabalho sempre que for observado necessário.

### 2.2.3 O Pilar adaptação

Segundo Schwaber e Sutherland (2013), a adaptação se dá no momento quando o *Scrum Team* e os *stakeholders* conhecem o sistema de desenvolvimento existente, 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 OS 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.

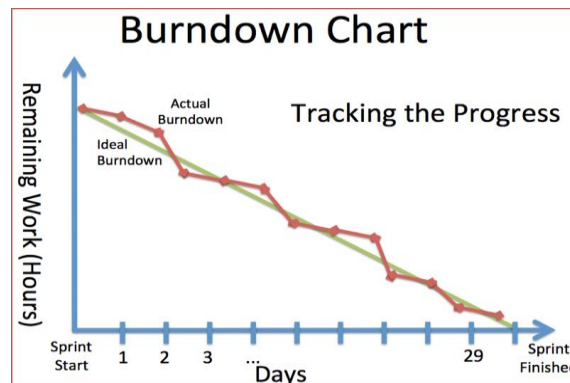
No *Sprint Backlog* para Sutherland e Schwaber (2013), estão atividades que o time mensura ter a capacidade se concluir durante a próxima *Sprint*. Os itens são movidos do *Product Backlog* para o *Sprint Backlog* de acordo com a prioridade estabelecida pelo *Product Owner* e a percepção do Time Scrum do 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.

### 2.3.4 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, (2013) <sup>2</sup>

Para Sutherland e Schwaber (2013), o gráfico de *Burndown* permite um fácil 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 OS EVENTOS

### 2.4.1 A definição de *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”,

---

<sup>2</sup> Disponível em:

[https://www.halvorsen.blog/documents/programming/software\\_engineering/resources/resources/Scrum%20Overview.pdf](https://www.halvorsen.blog/documents/programming/software_engineering/resources/resources/Scrum%20Overview.pdf)

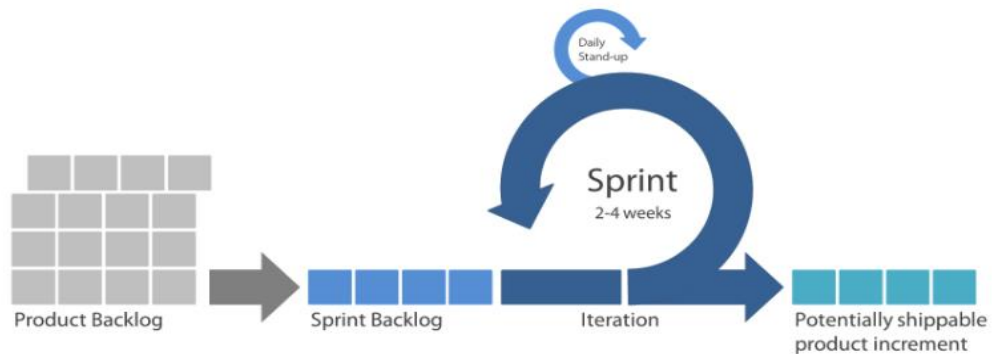
uma versão proveitosa, de preferência que já agregue valor ao usuário final, de produto que está em processo de concepção. As sprints através de todo esse processo criação, tem uma duração consistente no período de esforço de desenvolvimento. Uma nova sprint tem início logo após a finalização da sprint 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 Scrum* 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) <sup>3</sup>

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, enfrentamento 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.

---

<sup>3</sup> 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 saná-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 conhecerá 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 término 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.

### 3 METODOLOGIA SCRUM: IMPLEMENTAÇÃO

Schwaber e Sutherland (2013), observou que o Scrum, como uma metodologia ágil de desenvolvimento, possui dentre seus objetivos estabelecer uma comunicação mais ampla e assertiva e uma interação mais próxima entre os membros do time e o cliente, procurando uma maior transparência no processo de desenvolvimento e/ou possibilitar que o processo de adaptação de novas alterações ou requisitos estabelecidos pelo cliente sejam mais facilmente acolhidas. Porém, para que esses objetivos sejam alcançados, é necessário executar os passos descritos na metodologia Scrum que serão mencionados a seguir.

#### 3.1 OS PAPÉIS DA METODOLOGIA SCRUM

Segundo Schwaber e Sutherland (2013), uma das primeiras decisões a serem realizadas ao iniciar a implementação da metodologia Scrum em um projeto, é estipular quem se responsabilizará pela execução dos seguintes papéis que são necessários para essa metodologia. Esses papéis são: *Product Owner*, *Scrum Team* e *Scrum Master*.

##### 3.1.1 O *product owner*

Em Schwaber e Sutherland (2013), está descrito que esse papel, acaba sendo exercido em sua grande maioria de vezes pelo cliente em si, ou seja, a pessoa a qual o produto será entregue ao final do desenvolvimento. Este papel também pode ser executado por um representante direto ou até mesmo por um colaborador de dentro da própria empresa de desenvolvimento, desde que ela possua uma comunicação direta com o cliente final e tenha conhecimento suficiente sobre o que se deseja alcançar com o desenvolvimento do produto.

O *Product Owner*, também chamado de “P.O.” por Schwaber e Sutherland (2013), é o responsável por definir quais serão os itens que deverão ser confeccionados para compor o *Product Backlog* e também definir o grau de prioridade desses itens em questão, pois é ele quem deve possuir a visão geral do que os usuários do produto necessitam. Para o sucesso do *Product Owner*, todos os envolvidos devem respeitar suas decisões, e estas são visíveis no conteúdo e na priorização do Backlog do produto feita por ele.

Schwaber e Sutherland (2013), continua e ressalta que, durante todos os ciclos do processo de desenvolvimento é o *Product Owner* quem tem a palavra final sobre o que foi realizado no desenvolvimento. É o *Product Owner* quem avalia se o item que lhe foi entregue atende ou não aos requisitos que produto necessita. Tal responsabilidade lhe é designada, pois dentro do processo do Scrum, quem tem mais informações sobre as regras de negócio, os requisitos e sobre o que os usuários finais do produto necessitam é o *Product Owner*.

### 3.1.2 O *Scrum master*

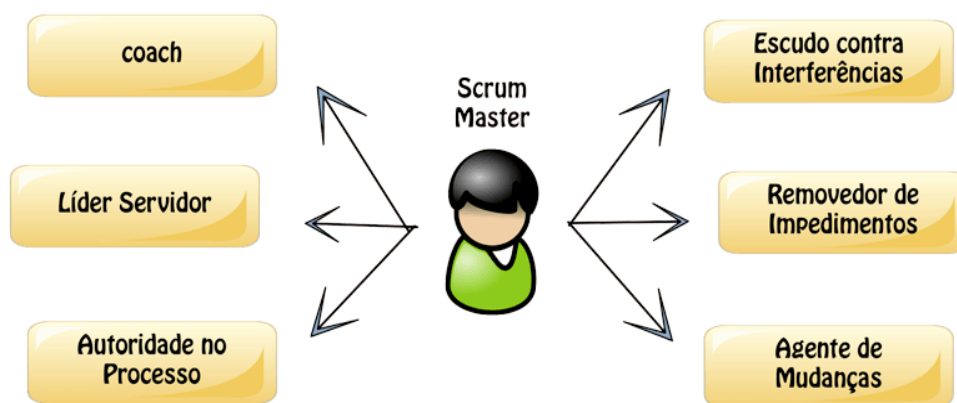
Para Schwaber e Sutherland (2013), o responsável para desempenhar o papel de *Scrum Master* deve ser um indivíduo que domina o processo do Scrum, pois este indivíduo será o encarregado por garantir que todo o time siga os princípios e fundamentos do Scrum, tornando-se uma autoridade neste processo. O *Scrum Master* é também incumbido de garantir que o *Scrum Team* não se responsabilize de forma excessiva com aquilo que ele pode produzir durante uma *Sprint*, dessa forma garantindo que o escopo acordado para a *Sprint* seja entregue, e que desse modo que não interfira de uma maneira negativa a qualidade do produto entregue.

Vieira (2014), diz que este papel, o *Scrum Master*, dentro da metodologia Scrum deve garantir que todo o time conheça os valores, princípios e práticas, bem como as abordagens específicas do time Scrum, ajudando a melhorar o processo de

desenvolvimento para aumentar o valor agregado ao produto do negócio em questão.

Segundo Vieira (2014), o *Scrum Master* também deve ter uma postura de facilitador, visando sempre soluções rápidas e que sejam adaptativas para todos e quaisquer impedimentos que surgirem ao período de uma *Sprint* dessa forma melhorando a produtividade do *Scrum Team*.

**Figura 03 – Papéis do *Scrum Master***



**Fonte:** Vieira, (2014)<sup>4</sup>

Em Schwaber e Sutherland (2013), o *Scrum Master* deve ser um Líder Servidor, ou seja, um indivíduo com foco em ajudar o time para que assim, todo o time, possam ser mais produtivos e eficazes durante todo o processo de desenvolvimento do produto. A qualquer momento que um problema ou impedimento surgir perante o *Scrum Team*, e seja essa situação no sentido de quem pode e quem deve resolvê-lo, o *Scrum Master* deve ser aquele que ajuda o

---

<sup>4</sup> Disponível na Internet via: <http://www.mindmaster.com.br/scrum-master/>

*Scrum Team* a resolver, e não sanar a situação pelo time. Entretanto, se a situação em questão não puder ser resolvida pelo time, este é o momento que então o *Scrum Master* toma o problema como uma responsabilidade sua, e tomará ações para resolvê-lo.

### 3.1.3 O *Scrum team*

Segundo Schwaber e Sutherland (2013), *Scrum Team* é a time responsável por todo o desenvolvimento do produto, os indivíduos que formam esse time, são os encarregados por transformar todos os itens o que estão no *Sprint Backlog* em implementos e funcionalidades no produto e entregá-los durante uma *Sprint*. O *Scrum Team* é também o responsável por estimar de esforço necessário para executar esses itens que estão no *Sprint Backlog*, e quando possível, prever obstáculos que possam surgir durante essa execução, e já tomar ações para mitigar esses obstáculos.

Para Schwaber e Sutherland (2013), no *Scrum Team* não há títulos, cargos ou funções pré-estabelecidas. Todos os membros do time trabalham em conjunto para completar a execução dos itens no *Sprint Backlog*. Não deve nem mesmo haver uma equipe voltada apenas para testes ou análise, todos fazem isso de forma conjunta durante todo o processo, dessa forma nivelando o nível de conhecimento dos indivíduos de todo time em todas as áreas.

Vieira (2014), nos diz que, o *Scrum Team* deve trabalhar de uma maneira auto organizável, ou seja, o próprio time é quem define a forma da qual eles devem trabalhar, ou a maneira como ocorrerá a execução de itens do *Sprint Backlog* em funcionalidades e/ou implementos. Nem o próprio *Scrum Master* deve interferir neste processo. Entretanto, para que isso seja melhor transformado em uma equipe auto organizável, é preciso que o *Scrum Team* seja composto por indivíduos com diferentes habilidades, pois isso aumentará a chance de que o objetivo final de uma *Sprint* seja executado apenas pelo próprio time, pois, quando um time é formado por

integrantes onde eles possuam o mesmo tipo de conhecimento, há a possibilidade que em um dado momento da *Sprint*, que eles não tenham conhecimento necessário para executar uma dada tarefa do *Sprint Backlog* e repassar essa tarefa para outro time é aumentada.

Em Schwaber e Sutherland (2013), a entrega final obtida nesta maneira de trabalho auto organizável, se torna em um aumento na sinergia entre os membros do *Scrum Team*, e o aperfeiçoamento da eficiência e da eficácia do time de forma geral. Nada indica que a não possa haver a substituição nos indivíduos do *Scrum Team*. Porém, pode-se observar que o ganho de sinergia e produtividade adquirida anteriormente pela equipe através da auto-organização, acaba sendo reduzida. Dessa forma, deve-se ser cuidadoso quando for realizada a alteração na composição de um *Scrum Team*, almejando sempre alocar os indivíduos que já estão familiarizados com a metodologia e com o projeto ao qual serão alocados.

### 3.2 O MODELO CÍCLICO ITERATIVO DO SCRUM

Segundo Schwaber e Beedle (2002), depois de ser estabelecido as definições dos papéis entre os indivíduos envolvidos com o projeto, para que o processo com o Scrum se inicie, é fundamental que já exista uma visão do produto que deverá ser entregue. Após a definição do produto, inicia-se a montagem do *Product Backlog*, que pode ser interpretado pelo time como uma transcrição daquilo que deverá ser desenvolvido pelo o time, na forma de atividades e incrementos necessários para o completo funcionamento do produto. Se faz fundamental o entendimento que mesmo o *Product Backlog* sendo elaborado no início do projeto, a probabilidade indica que ele não estará completo. A fase inicial deve ser pensada como requisitos base para o processo de desenvolvimento do produto, e que a medida que o projeto vai evoluindo, o *Product Backlog* pode evoluir de maneira conjunta.

Para Schwaber e Beedle (2002), em seguida a este processo, começam as iterações existentes no Scrum, essas iterações são conhecidas como *Sprint*, e

devem possuir uma duração compatível com aquilo que deve ser executado neste período. Ao encerramento de cada Sprint é entregue um novo incremento do produto a ser lançado (levando em considerações as prioridades estabelecidas pelo *Product Owner* durante a confecção do *Product Backlog*), estes tendo passado pelos processos de desenvolvimento, aprovação e teste. A *Sprint* deve ser vista como o coração de todo o processo do Scrum, onde nela é que ocorre a interação entre o time, e neste processo de compartilhamento de conhecimento, superação de problemas e a obtenção do objetivo determinado pelo próprio time, faz-se o núcleo do Scrum.

Em Schwaber e Beedle (2002), é dito que durante as Sprints são efetuadas as *Dailys Scrum*, estabelecidas anteriormente como reuniões diárias de curta duração, aproximadamente de 10 a 15 minutos para sua realização, com o propósito de evidenciar aos próprios membros do *Scrum Team* o que está sendo executado, as dificuldades e o que ainda deverá ser feito. A *Daily Scrum* não deve ser vista como um momento para expor ao gerente quem está ou não atrasado, mas sim esclarecer o andamento de cada tarefa executada por um indivíduo do time dentro do projeto para o resto do time, dessa forma estabelecer compromissos entre os membros do time, sem o envolvimento de pessoas externa ao time.

Conforme Takeuchi e Nonaka (1986), todos os membros do time iniciam seus trabalhos acompanhando um mesmo ritmo, tanto individual quanto coletivamente, cria-se uma sinergia completamente nova, e esta sinergia acaba se tornando uma força motriz que faz com que o time se mova de forma evolutiva.

### 3.3 A ABORDAGEM DISTRIBUÍDA PARA O SCRUM

Em Oliveira e Lima (2011), existe um tópico que precisa ser considerado sobre a metodologia Scrum, que é a possibilidade da mesma ser aplicada em times que se encontram em pontos geograficamente distante, sejam esses pontos cidade/estado, e/ou até mesmo em países diferentes. A execução do Scrum em ambientes onde as equipes se encontram em pontos remotos, pressupõem em



adaptações com o uso de ferramentas de comunicação e conferência para que as reuniões do Scrum aconteçam.

Entretanto, foi observado por Prado e Ferrari (2013), que mesmo empregando ferramentas de comunicação para cumprir as reuniões previstas dentro da metodologia Scrum, problemas ainda acabam sendo vistos quando esta metodologia é aplicada em um ambiente de Desenvolvimento Distribuído de Software (DDS). O gerenciamento das equipes distribuídas, e a gestão das atividades, e a falta de domínio na plataforma utilizada, e as ferramentas de comunicação, reuniões diárias e integração, foram dificuldades enxergada durante os projetos abordados durante a pesquisa realizada por Prado e Ferrari (2013), sendo um deles com um time formado por 6 indivíduos utilizando o sistema de Scrum comum e outro time formada por 50 indivíduos, fazendo o uso do sistema de “Scrum de scrums”.

Segundo Sutherland (2001), o *Scrum of scrums* é uma técnica para escalar o Scrum para grandes grupos (dezenas de pessoas), consistindo em dividir os grupos em times ágeis de 5 - 10 pessoas. Cada Daily Scrum realizada com um desses times, termina designando um dos membros como “embaixador” para participar de um encontro diário com os embaixadores dos outros times, chamado de Scrum de scrums.

De acordo com Possa (2013), apesar desse método trazer inúmeras dificuldades para a aplicação da metodologia dentro de um time maior e/ou distantes geograficamente, a capacidade de resposta às mudanças de requisitos, o aumento na produtividade e a maior satisfação das pessoas na área de negócios acabam sendo pontos positivos que fazem com que empresas aceitem esse este tipo de organização.

### 3.3.1 O *Scrum of scrums*

Segundo Paasivaara, Lassenius e Heikkila (2012), para que este método se torne efetivo, se faz necessário que todo o processo seja revisado e adaptado para

que este se encaixe de maneira mais proveitosa para todo o time. Entre essas adaptações está o *Scrum of scrums*, que é em uma reunião com representantes de cada time, que tem um formato extremamente parecido com a Daily Scrum, entretanto, cada membro responderá a outras perguntas. São elas:

- O que a sua equipe fez desde a última reunião que possa vir a ser importante para outra equipe?
- O que a sua equipe irá fazer a seguir que possa ser relevante para outra equipe?
- Existe impedimento no caminho da sua equipe que possa vir a atrapalhar?

De acordo com Paasivaara, Lassenius e Heikkila (2012), existem relatos sobre os benefícios de sua utilização em projetos que visam o escalonamento da metodologia, o uso Scrum of scrums pode acabar ocorrendo de forma extremamente errada e tornando-se até mesmo uma perda de tempo, exigindo dos seus participantes transparência e maturidade sobre a metodologia.

## **4 METODOLOGIA SCRUM: UMA PERSPECTIVA GLOBAL**

No dado momento, é possível possuir uma perspectiva de como o Scrum funciona dentro de um projeto, de suas características e de seus objetivos quando a sua implementação é realizada de maneira correta. Entretanto, se faz necessário estabelecer pontos sobre essa metodologia de desenvolvimento ágil para que se tenha a lucidez de que talvez, existe casos, que ela não seja a metodologia mais apropriada para tipos de empresas ou até mesmo grupo de indivíduos.

### **4.1 AS VANTAGENS INERENTES AO SCRUM**

Para Schwaber e Beedle (2002), a metodologia Scrum visa valorizar os indivíduos alocados no projeto, prezando pelo estímulo de interações, para que todos estejam conscientes dos passos, das alterações que possam vir a ocorrer, cronogramas que devem ser seguidos e das atividades a serem estabelecidas e executadas, gerando um maior ligação e comprometimento por parte dos indivíduos que fazem parte do time para com o projeto. Segue as vantagens identificadas no processo de pesquisa.

Schwaber e Beedle (2002), verifica um maior comprometimento por parte do time, já que o time estará ciente dos passos a serem executados durante o projeto devido a transparência da metodologia.

Schwaber e Sutherland (2013), indica uma flexibilidade em relação a redefinição de prioridades, uma vez que nem sempre o cliente terá todas essas definições, e mesmo quando houver essas definições, as mesmas podem ter alterações de prioridade devido a motivos durante a execução do projeto. Nestas situações, o Scrum tem uma característica flexível para definir uma nova prioridade ou alterar uma prioridade já existente.

Ferreira (2005), aponta uma considerável redução na parcela de bugs gerados. Devido ao time estar alocado em todo ciclo de produção desde análise de

requisitos, implementação e teste, isto gera uma maior lucidez sobre o negócio em questão contribuindo diretamente na qualidade e funcionalidade do produto, e tendo este foco tanto em qualidade quanto a funcionalidade, a quantidade de bugs e de retrabalho é reduzida de maneira considerável.

Schwaber e Beedle (2002), descreve uma perspectiva para que todos os envolvidos possam visualizar o projeto como um todo. As atividades que devem ser executadas para a conclusão do projeto são organizadas no *Product Backlog*, esse é um artefato onde todos devem possuir acesso de maneira fácil, para que todos os indivíduos estejam atualizados sobre andamento corrente e futuro do projeto.

Schwaber e Sutherland (2013), mostra que a existência de curtos intervalos entre as entregas do produto, as *Sprints*, para que o cliente possa fazer acompanhamento de perto do progresso do produto e fornecer um feedback contínuo sobre as funcionalidades do produto para o time.

Para Ferreira (2005), a possibilidade de uso de metodologias em paralelo, uma vez que o Scrum permite ser implementado em conjunto com outras metodologias, para dessa forma conseguir suprir lacunas que possam ser encontradas dentro da própria metodologia Scrum.

## 4.2 AS POSSÍVEIS DESVANTAGENS NO SCRUM

Schwaber e Beedle (2002), diz que justamente por o Scrum possuir seu foco voltado para a qualidade e funcionalidade do produto, e para isso busca um maior comprometimento e motivação do time, essas prioridades permitem ao Scrum gerar brechas para o surgimento de pontos negativos, e esses pontos devem ser considerados antes de sua implementação. Segue as possíveis desvantagens identificadas no processo de pesquisa.

Schwaber e Beedle (2002), aponta que o Scrum ao ter entre suas prioridades uma maior qualidade ao invés de foco na entrega, os prazos podem acabar não

sendo contemplados de maneira devida, e isso reverter se em atrasos na entrega do produto.

Para Ferreira (2005), existe uma falta de documentação presente no Scrum. Pelo seu foco estar voltado na agilidade e buscar sempre atender as necessidades cliente, pode ocorrer de apenas documentar os pontos mais críticos do projeto, permitindo que todo o resto não seja documentado de maneira devida, o que pode gerar problemas na necessidade de analisar um questão específica do projeto, neste momento verificasse a falta de documentação sobre a questão.

Schwaber e Sutherland (2013), faz referência sobre a possível confusão que pode haver sobre o papel de cada indivíduo no projeto. Exatamente por possuir determinadas funções indefinidas, o time pode ter problemas com os processos de comunicação interna e/ou dúvidas sobre suas tarefas.

Ferreira (2005), relata que o fato de deixar com que o próprio time tomar decisões referentes aos seus processos internos, pode acabar resultando, em sua grande maioria de vezes ao início de um projeto Scrum, em resoluções equivocadas que consequentemente irão resultar em falhas do *Scrum Team*, mas que podem ser evitadas com a intervenção de um membro externo ao time com conhecimento sobre a resolução em questão.

Ferreira (2005), verificou que em casos onde o time já está acostumado com a implementação de uma forma convencional de gestão de projetos, a primeira impressão que eles formarão em relação ao Scrum pode não ser muito positiva, é necessário haver um período de adaptação para este time. Após este período de adaptação, ainda é capaz que membros do time ainda não se sintam confortáveis e receptivos as obrigações que o Scrum exige.

Schwaber e Sutherland (2013), deixa claro que o Scrum é tem um melhor aproveitamento em pequenas equipes, compostas de 4 a 10 pessoas, pois em times maiores que isso pode resultar em algo confuso tanto para o *Scrum Team*, bem como para o *Product Owner* e *Scrum Master*. Time maiores que quantidade sugerida, podem ser subdivididos em times menores, porém isso acaba impactando de forma negativamente na rotina do *Product Owner* e *Scrum Master*, nesta situação o *Product Owner* e *Scrum Master* podem ser sobrecarregados.

### 4.3 OS ASPECTOS DA TRANSPARÊNCIA NO SCRUM

É dito por Schwaber e Sutherland (2013), que a transparência não ocorre de maneira fácil, mas é um caminho que deve ser percorrido com esforço.

Segundo Carvalho e Mello (2012), existe uma necessidade que ao início do projeto, para certas organizações, que ocorra a quebra de limites hierárquicos e melhoria de relacionamento dentro do time para que todos os indivíduos se sintam parte do time, dessa forma cada membro possuirá as suas responsabilidades e que estejam no mesmo nível. Para isso, é considerado uma boa prática que os membros do time se intitulem apenas pelo primeiro nome sem tratamento especial para facilitar este processo.

Ainda Carvalho e Mello (2012), constatou que a transparência entre o *Scrum Master*, o *Product Owner* e o *Scrum Team* são indispensáveis, junto a uma relação de sinceridade e um canal comunicação permanente que seja utilizado por todos eles. Isto se torna em um ponto extremamente vital dentro da metodologia Scrum, visto que a comunicação é um dos pontos mais importante para seu funcionamento íntegro, a transparência acaba sendo algo de suma importância pois é através dela que, ocorre uma diminuição do risco de problemas ou retrabalho durante o processo de desenvolvimento do produto.

Schwaber e Sutherland (2013), lembra que os artefatos do Scrum (*Product Backlog* e *Sprint Backlog*) devem ter fácil acesso por todos os indivíduos do time, e seu conteúdo deve ter fácil entendimento para que desta forma, todas as decisões que forem tomadas durante os processos, contenham uma base sólida, para haver uma mitigação nas chances de falha.

É dito por Schwaber e Beedle (2002), que para essa transparência seja obtida, se faz necessário que os indivíduos que compõem o *Scrum Team* bem como o *Scrum Master*, trabalhem como um time unido, e amadureçam o processo e iniciem o estabelecimento do próprio ritmo de trabalho do time.

#### 4.3.1 O conceito “Pronto”, sua definição

Schwaber e Sutherland (2013), relata que quando há a aplicação de uma metodologia em um projeto, seja esta metodologia escolhida o Scrum ou não, existe um desejo de maior transparência durante as etapas deste projeto. E neste caminho em encontro a transparência, existe um ponto a ser estabelecido inicialmente pelo time e que deve ser amadurecido com o transcorrer deste projeto, que é a definição de “Pronto”, em inglês *DoD(Definition of Done)*.

Para Parente (2015), existem diferentes definições conferidas ao conceito de “pronto” em diferentes textos e artigos por diferentes autores. Ainda em Parente (2015), a definição de “Pronto” são enumeras validações ou checagens que um time deve realizar, sendo ou não Scrum, antes de realizar a entrega de produto, ou parte dele. Esta verificação deve assegurar que produto em questão está atendendo os requisitos solicitadas, tanto funcionais ou não funcionais, com o foco em atender as reais necessidades do cliente.

Schwaber e Sutherland (2013), no *Scrum Guide* escolhem, e de maneira consciente, não expor uma definição sobre o que é “Pronto” perante os seus pontos de vista, decidindo por responsabilizar o *Scrum Team* para esta definição, o que é “Estar Pronto”, cientes que para distintos *Scrum Teams* esta definição pode ser totalmente diferente. Em diferentes momentos, “Pronto” pode significar que uma nova funcionalidade foi desenvolvida, testada e aprovada, enquanto em outro momento “Pronto” pode representar que uma certa funcionalidade foi apenas codificada pelo time.

Schwaber e Sutherland (2013), continua e diz que independente de qual será a definição de “Pronto” estabelecida pelo time, se faz necessário que a esta seja coesa, e que todos os indivíduos que formam o time, tenham o conhecimento claro para aferir quando algo “Está Pronto”, para que dessa forma em momentos futuros não haver erros quanto a este conceito dentro do time. Com a possibilidade de todos os times gerarem a sua própria definição de “Pronto”, se espera que proveitos comecem a surgir por todo o processo de desenvolvimento do produto, uma vez que não mais haverá incertezas de que algo está ou não “Pronto”.

Vashishtha (2014), alerta que uma definição coerente para “Pronto”, tende a auxiliar o *Product Owner* na organização durante todo o projeto e ao mesmo tempo em que colabora a todos os indivíduos do time a ter ciência do momento em que um bloqueio ocorreu em suas atividade, dessa maneira permitindo ações para que o bloqueio seja sanado, gerando uma maior agilidade em toda a *Sprint*.

#### 4.3.2 A estimativas de atividades e suas dificuldades

Para Braz (2011), no interior do Scrum é de extrema relevância que as tarefas que ainda serão executadas sejam estimadas de forma assertiva, para se proteger de eventuais atrasos durante o caminho. Seguindo esse pensamento, atribui-se 4 pontos em que acabam sendo gerando transtornos no processo de planejamento de um projeto.

O primeiro ponto para Braz (2011), é presumir que as tarefas a serem executadas são independentes: É muito comum uma tarefa depender de outra para ser concluída. Dessa maneira caso uma tarefa tenha sua estimativa equivocada, isto impactará diretamente em todas as outras tarefas que possuem dependência dela. Neste cenário se faz necessário que ao levantarmos novos requisitos para uma determinada atividade, deve se atualizar sua estimativa em conjunto.

Braz (2011), continua seu relato e o segundo ponto é que, atrasos impactam o cronograma, antecipações não. É necessário que exista um planejamento sobre qual tarefa deve ser iniciada, após o término de outra. Se uma tarefa atrasou, é imprescindível finalizá-la antes de iniciar outra, e apenas adiantar uma tarefa se e somente se, as tarefas anteriores forem concluídas antes do prazo estipulado e haja os recursos, humanos ou não, para poder adiantá-la.

O próximo ponto para Braz (2011), é a “síndrome do estudante”, está se resume em começar a execução de uma atividade no último momento hábil, que não interferirá na entrega do produto. Se existe uma tarefa foi estimada para uma determinada número de dias para a sua execução, e sabe-se que existe a possibilidade de ser terminada antes, é recomendado começar o seu



desenvolvimento já no prazo estipulado, para caso ocorra uma situação imprevista, o tempo restante servirá para resolvê-lo em tempo hábil.

E o último ponto para Braz (2011), é executar atividades em modo multitarefas. É necessário evitar executar mais de uma atividade por vez, independentemente do quão simples ela possa parecer ou ser. Neste cenário, é recomendado produzir o fluxo de trabalho o mais ágil possível, pois se existir a necessidade de passar a atividade de um indivíduo A para um indivíduo B durante o processo de desenvolvimento, isto se torna em algo mais simples caso o indivíduo A possua apenas uma atividade atribuída para ele. No momento em que um indivíduo tenta executar mais atividades em paralelo, seu desempenho é afetado de maneira negativa em todas.

Mesmo com todas essas situações adversas, Alexandre Vignado(2016), mostra uma maneira de estimar atividades dentro do Scrum de forma dinâmica, sem gerar foco na questão crucial de “quantas horas/dias são suficientes para desenvolver essa funcionalidade”, já que provavelmente o time irá se lembrar principalmente de “O time possui X dias”, e não o contrário, “X dias seriam ideais para o time”. Esse método de estimativa é conhecido como *Planning Poker*.

#### 4.3.2.1 O *planning poker* e seu funcionamento

Kniberg (2007), recomenda que o *Planning Poker* deve ser realizado na reunião de definição da *Sprint*, o momento onde as estimativas para as funcionalidades que serão executadas na *Sprint* serão realizadas. Para começar o *Planning Poker*, é preciso que o time tenha em mãos as funcionalidades que necessitam ser concluídas pelo time na *Sprint* que estão planejando. Dessa forma, cada membro do time estipula um valor para aquela funcionalidade levando em conta a sua complexidade, e a quantidade de tempo que ele acredita ser necessário para completar a funcionalidade, levando em conta o conceito de “Pronto” definido pelo próprio time anteriormente. Assim sendo, cada membro do time possui um baralho formado por 13 cartas, cada uma representando os seguintes valores: 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40 e 100, interrogação (que demonstra que o membro do time não foi

capaz de estimar a funcionalidade em questão, e a xícara de café, que sugere uma pausa no processo de estimativa.

Com todos os membros do time munidos de seus baralhos, Kniberg (2007), continua e relata que as funcionalidades devem apresentadas uma-a-uma e todo o time estipula uma carta para a funcionalidade que está sendo apreciada no momento. Em valores acima de 20, que são os considerados como altos, significa que aquela funcionalidade não conseguirá ser concluída em apenas uma entrega devendo ser quebrada em outras pequenas atividades ou sub funcionalidades. O ideal é que, ao final da reunião de planejamento, as atividades elaboradas para uma determinada funcionalidade sejam estimadas entre  $\frac{1}{2}$  e 13, mas permitir que uma funcionalidade considerada grande, acabe se tornando extremamente dividida, podendo fazer com que o time seja penalizado com o processo de micro gerenciamento.

Segundo Kniberg (2007), caso ocorra durante a estipulação de cartas discordância entre os valores definidos pelos membros do time, todos os membros deverão realizar uma conversa com questionamentos sobre esta atividade que possam chegar a um consenso sobre a sua complexidade para sua execução, para que dessa forma todos possam entender melhor o que precisa ser realizados sobre esta atividade, é percebido que quando uma nota considerada baixa é escolhida por um membro do time, existe a possibilidade que certas complexidades não estejam sendo observadas por esse membro, bem como pode acontecer o inverso de uma tarefa estar sendo envergada com maior complexidade do que de fato terá.

Haugen (2006), notou um progresso na eficiência em estimar funcionalidades com esse método de atribuição de pontos. Em contra ponto, Alves (2012) ressalta que um ponto deve ser considerado durante este processo, é que os indivíduos que possuem maior entendimento sobre o que deve ser executado na atividade em análise, tendem a exercer grande influência na estimativa a ser proposta por todo o time, desta maneira os membros que se deixam influenciar seguem aquilo que o membro com maior conhecimento sobre atividade acredita, e deixam de expor aquilo que realmente pensam.

Alves (2012), relata um software criado pela *Mountain Goat Software*, cujo o nome também é *Planning Poker*, com intuito de uso do *Planning Poker* de forma virtual e remota, com a necessidade de um cadastro pelos membros do time, compartilhamento do link para sessão pelo criador da sala, e elencar quais as atividades deverão ser estimadas.

Haugen (2006), aponta *Planning Poker* como uma excelente alternativa para times que estejam lidando com dificuldades no processo de estimar funcionalidades e atividades, ou que apenas procurem algo mais interativo, ágil e que estimule o nivelamento de conhecimento entre os membros do time. O uso deste método de forma tradicional em equipes geograficamente distantes se torna inviável, visto que o time não compartilha do um mesmo espaço físico. Mas existem alternativas para isso, com uso de ambiente virtual online.

## 5 CONSIDERAÇÕES FINAIS

O mercado de desenvolvimento de software, e praticamente todo e qualquer modelo empresarial, busca maneiras de capacitar sua organização, tanto internamente quanto externamente, o mais eficiente e eficaz possível, possibilitando uma maior qualidade em seus produtos e serviços, conseqüente gerando maiores lucros e promovendo a imagem da organização como um todo. Incorporadas a este contexto, as metodologias ágeis, dentro destas metodologias o Scrum ganha destaque, e vem adquirindo um espaço mais considerável dentro das estruturas organizacionais nas empresas.

Dentre essas metodologia ágeis, o Scrum se apresenta como uma metodologia que anseia por uma maior transparência entre o time de desenvolvimento de software e o cliente, auxiliando desde a definição de requisitos até a entrega final do produto, com o intuito de otimizar processos e entregar um produto de qualidade que atenda aos requisitos e as reais necessidades do cliente. Pelo meio da confecção do *Product Backlog*, definições dos conteúdos que irão gerar as *Sprints*, realização das reuniões diárias conhecidas como *Daily*, e outras reuniões sejam elas semanais e mensais necessárias entre o time, em conjunto de entregas constantes do produto, junto ao cliente, dessa maneira o cliente fica constantemente a par do que está sendo desenvolvido para o produto, e como anda o processo de desenvolvimento geral do produto.

Já analisando o seu funcionamento interno, o Scrum determina que o time de desenvolvimento do produto deve procurar estar em sintonia e tratar a todos os membros e indivíduos ligados ao processo com transparência, relatando as dificuldades que estão sendo verificadas durante o percurso, e para que possam saná-las de forma conjunta quando necessário. Cargos e papéis devem ser irrelevantes dentro do time e todos devem trabalhar como iguais, com um mesmo objetivo, aumentando o comprometimento do time.

Entretanto, mesmo possuindo um foco em aceleração do processo de desenvolvimento do produto de software, o Scrum acaba sendo uma alternativa arriscada para grandes empresas, ou empresas com o processo “engessado”, visto

que seu processo de implementação pode se tornar complexo, podendo modificar em cenários específicos, quase que totalmente, a estrutura organizacional na empresa na qual será implementado. Havendo a possibilidade de causar transtorno sobre o papel de cada indivíduo no time.

Na realização desta pesquisa, ficou evidenciado que o Scrum possui uma melhor adequação para pequenos times, sendo crucial que um estudo aprofundado e prévio da organização a qual existe a intenção da sua implementação, para analisar se esta metodologia ágil irá suprir de maneira positiva às reais necessidades do time e da organização em geral, e por consequência seus clientes e consumidores.

## REFERÊNCIAS

ALVES, Dennis. Ferramenta para apoio à estimativa baseada em Planning Poker utilizando a metodologia Scrum. Recife. 2012. Disponível em: <<http://www.cin.ufpe.br/~tg/2012-1/dwas.pdf>> Acesso em: 03 outubro 2018.

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

BRAZ, Alan. Introdução ao Scrum. [S.l.]. Disponível em: <<http://www.ic.unicamp.br/~ariadne/mc436/1s2014/Scrum-Alan.pdf>> Acesso em: 03 outubro 2018.

CARVALHO, Bernardo; MELLO, Carlos. Aplicação do método ágil Scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica São Carlos. Gestão e Produção, 2012: v. 19 n.3 . Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0104-530X2012000300009](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X2012000300009)>Acesso em: 03 outubro 2018.

FERREIRA, Décio. et al. SCRUM: Um modelo ágil para gestão de projetos de software. Disponível em: <[http://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es\\_final\\_19.pdf](http://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es_final_19.pdf)>. Acesso em: 02 outubro 2018.

Haugen, N. C. (2006). "An Empirical Study of Using Planning Poker for User Story Estimation". Proceedings of the Conference on AGILE 2006 (pp. 23-34). Washington (DC): IEEE Computer Society. Disponível em: <<https://ieeexplore.ieee.org/document/1667560>> Acesso em 01 outubro 2018.

KNIBERG, Henrik. Scrum and XP from the Trenches: How we do Scrum. [S.l]

InfoQ. 2007. Disponível em: <<https://www.infoq.com/minibooks/scrum-xp-from-the-trenches-2>> Acesso em: 04 setembro 2018.

LIMA, Rosangela; OLIVEIRA, Eneida. Estado da arte sobre o uso do Scrum em ambientes de desenvolvimento distribuído de software. Revista de Sistemas e Computação. 2011, v. 1, n.2: p. 106-119. Disponível em: <<http://www.revistas.unifacs.br/index.php/rsc/article/view/1902>>. Acesso em: 01 outubro 2018.

PAASIVAARA, Maria; LASSENIUS, Casper; HEIKILLA, Ville. Inter-team Coordination in Large Scale Globally Distributed Scrum: Do Scrum-of-Scrums Really Work?, ESEM '12 Proceedings of the ACM-IEEE ISESE P. 235-238. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2372251.2372294>>. Acesso em: 01 outubro 2018.

PARENTE, Herbert. Definição de Pronto, 2015. Disponível em: <<http://governoagil.com.br/2015/05/25/definicao-de-pronto/>>. Acesso em: 04 setembro 2018.

PRADO; FERRARI. Avaliação do Impacto do Desenvolvimento Distribuído de Software em um Projeto Adotando o Scrum: Um Estudo Comparativo, 2013, v. 2, n.3: p. 201-210 Disponível em: <<http://revistatis.dc.ufscar.br/index.php/revista/article/download/64/59>>. Acesso em: 01 outubro 2018.

POSSA, João. Escalonando o Scrum Dentro de uma Empresa de Desenvolvimento de Software com Equipes Geograficamente Distribuídas, 2013. Brasília, p. 46. Disponível em: <<http://bdm.unb.br/handle/10483/13611>> Acesso em: 01 outubro 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](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 Product Development Game. Harvard Business Review 64, no. 1, 1986.



VASHISHTHA, SHRIKANT. Definition of Ready, Disponível em: <<http://www.agilebuddha.com/agile/improve-sprint-throughput-with-definition-of-ready/>> Acesso em: 03 outubro 2018.

VIEIRA, Denisson. Scrum: A Metodologia Ágil explicada de forma definitiva. 2014.

VIGNADO, Alexandre. Afinal, como estimar usando o Planning Poker? 2016. Disponível em: <<http://www.alefininho.com.br/xk/?s=planning+poker/>> Acesso em: 03 outubro 2018.