

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO



CURSO DE ESPECIALIZAÇÃO EM ENGENHARIA DE SOFTWARE

CAMILA MESTIERI MARAFIOTI

**Utilizando Sprints de Scrum para aquisição de
conhecimento: refinamento de requisitos e melhoria
da modelagem em projetos *Big Data***

SÃO PAULO, MAIO/2018

CAMILA MESTIERI MARAFIOTI

Utilizando Sprints de Scrum para aquisição de conhecimento: refinamento de requisitos e melhoria da modelagem em projetos *Big Data*

Monografia apresentada ao Curso de Especialização em Engenharia de Software da Pontifícia Universidade Católica de São Paulo, como requisito parcial para obtenção do título de Especialista em Engenharia de Software, orientado pelo Prof. Dr. Italo Santiago Vega

SÃO PAULO, MAIO/2018

À minha família, que sempre me apoiou e incentivou ao longo desses anos.

AGRADECIMENTOS

Nesta gratificante etapa acadêmica, gostaria de agradecer a todos que colaboraram para o meu crescimento acadêmico e profissional ao longo desses anos.

Agradeço ao professor Italo Santiago Vega por ter aceitado o convite de me orientar, por ter me incentivado a aprimorar cada vez mais a pesquisa e pela confiança neste trabalho.

Agradeço aos colegas e amigos da turma de Engenharia de Software de 2017 da PUC-SP por todos os momentos e a todos aqueles que de alguma forma contribuíram ou torceram pela realização desta pesquisa.

Por último, mas não menos importante, agradeço aos meus colegas de trabalho da TIM que colaboraram direta ou indiretamente para a realização deste trabalho.

“Às vezes, quando você inova, comete erros. É melhor admiti-los rapidamente e continuar a melhorar suas outras inovações.” (Steve Jobs)

RESUMO

A evolução da tecnologia impactou a dissipação da informação e conseqüentemente provocou um aumento de geração de dados. E com isso, as empresas precisam, cada vez mais, serem mais ágeis e tomarem melhores decisões. Portanto, neste trabalho são apresentados o conceito e funcionalidades do *framework* Scrum e as principais etapas de um projeto *Big Data*. Para se atingir os objetivos do estudo, foi criado um processo de Sprint de aprendizagem para projetos *Big Data* utilizando Scrum. Ou seja, a partir do conceito do *framework* Scrum e processos de projetos *Big Data*, o Scrum foi adaptado para utilizar as Sprints para aprendizagem de *Big Data* e com isso poder tornar a equipe de projeto mais heterogênea hierarquicamente.

Palavras-chave: Scrum, *Big Data*, Sprints de aprendizagem.

ABSTRACT

The evolution of the technology has impacted the information dissipation and consequently caused an increase of data generation. And thereby, companies need, increasingly, to be more agile and make better decisions. Therefore, this paper presents the concept and functionalities of the Scrum framework and the main steps of a *Big Data* project. To achieve the study objectives, a Sprint learning process was created for *Big Data* projects using Scrum. That is, from the concept of the Scrum framework and *Big Data* project processes, Scrum has been adapted to use the Sprints for learning *Big Data* and thus can make the project team more heterogeneous hierarchically.

Keywords: Scrum, *Big Data*, Learning Sprints

LISTA DE FIGURAS

Figura 1 – Exemplo de Descrição da Sprint de Aprendizagem no Dicionário de Conhecimento	34
Figura 2 – Exemplo de Dicionário de Conhecimento - Item do Backlog.....	35
Figura 3 – Exemplo Quadro de Dúvidas	37
Figura 4 – Exemplo de comparativo gráfico das Sprints - antes, durante e depois da utilização de Sprints de aprendizagem.....	39

LISTA DE TABELAS

Tabela 1 – Resumo Scrum.....	30
Tabela 2 – Resumo Projeto <i>Big Data</i>	31
Tabela 3 – Resumo Processo	40

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Motivação	11
1.2	Objetivo	13
1.3	Método de trabalho	13
1.4	Organização do texto	13
2	Big Data e dificuldades em projetos de TI	15
2.1	Definição de <i>Big Data</i>	15
2.2	Projeto <i>Big Data</i>	17
3	Scrum	20
3.1	Métodos ágeis e solução de problemas	20
3.2	Deficiências do papel Development Team (Time de Desenvolvimento)	21
3.2.1	Product Owner (Dono do Produto)	21
3.2.2	Scrum Master	22
3.2.3	Development Team (Time de Desenvolvimento)	22
3.3	Engrenagens que movimentam o Scrum	23
3.3.1	Product Backlog	23
3.3.2	Sprint Backlog	24
3.3.3	Incremento do Produto	25
3.4	Eventos	25
3.4.1	Sprint	26
3.4.2	Sprint Planning (Planejamento do Sprint)	26
3.4.3	Daily Scrum (Reunião Diária)	27
3.4.4	Sprint Review (Revisão do Sprint)	27
3.4.5	Sprint Retrospective (Retrospectiva do Sprint)	28
4	Scrum e Sprints para aprendizagem de projetos <i>Big Data</i>	29
4.1	Processo	29
4.1.1	Definição de Papéis	31
4.1.2	Artefatos e Eventos	32
4.1.3	Comparativo gráfico	38
4.1.4	Resumo do processo final	39
5	CONSIDERAÇÕES FINAIS	41

REFERÊNCIAS.....	44
------------------	----

1 INTRODUÇÃO

Este capítulo aborda o início da problemática levantada sobre a utilização de metodologias ágeis. Elas têm sido mencionadas como uma excelente alternativa para projetos que possuem muitas mudanças, isto é, onde os requisitos são alterados frequentemente, como por exemplo, projetos *Big Data*. Porém, é esquecido que nestas situações, as equipes são pequenas e normalmente formadas por especialistas. Logo, os problemas encontrados no conhecimento da engenharia de *software* para *Big Data* não são solucionados.

Neste capítulo são descritos os elementos principais que definem esta pesquisa: Motivação, Objetivos, Método de Trabalho e Organização do Texto.

1.1 Motivação

Desde sempre a informação está presente no cotidiano das pessoas, porém, no decorrer dos anos, para que sua transmissão fosse mais rápida e abrangente, foram criados vários veículos de comunicação, entre eles: o jornal, o rádio, o telefone, o computador e o smartphone. Logo, pode-se dizer que a tecnologia revolucionou a forma como as pessoas interagem, pois, qualquer acontecimento ocorrido em qualquer parte do mundo é divulgado quase que instantaneamente nos dias atuais.

Diante do crescimento da quantidade de usuários conectados em seus *smartphones*, em redes sociais e na *internet*, houve um aumento na geração de dados não estruturados, isto é, postagens em redes sociais, vídeos, fotos, músicas, entre outros. A geração de dados não estruturados impactou o mundo empresarial e a economia devido aos novos segmentos de mercados, modelos de negócios e necessidade de acompanhamento de tendências.

Portanto, neste mundo globalizado e conectado, nasceu uma nova fonte de poder que não se pode ignorar, este imenso volume de dados que impacta os negócios e pode ser analisado para melhores tomadas de decisões, o *Big Data*. Logo, para que um negócio prospere, os acontecimentos do mundo devem ser

levados em consideração, fazendo com que as empresas se tornem mais ágeis para conseguir se adaptar as possíveis mudanças.

Em paralelo a esta evolução, ocorreu também a evolução das metodologias de processo desenvolvimento de *software*. De acordo com Soares (2004), até o início da década de 90, o modelo clássico (cascata/linear) dominava o modo de desenvolvimento de *software* nas empresas, porém, em 2001, o termo “metodologia ágil” se tornou popular. Neste mesmo ano, foi criado o manifesto ágil, que descreve princípios que fundamentam o desenvolvimento ágil de *software*. Após sua criação, novas metodologias de desenvolvimento de *software* foram chamadas de métodos ágeis.

Um destes métodos ágeis criados foi o Scrum, e segundo Rubin (2012), ele é uma abordagem ágil para desenvolvimento de *softwares*, e também é um *framework* utilizado para gerenciar o desenvolvimento de sistemas complexos desde o início da década de 1990.

Diante desta realidade, cada vez mais aumenta a importância de transformar dados em informação, isto é, a realização de projetos *Big Data* eficazes e com um gerenciamento de projetos que tenha um tempo de resposta que atenda às exigências do mercado, pois atrasos no desenvolvimento destes projetos podem comprometer as estratégias empresariais e podem limitar a agilidade que a empresa precisa para viver em um local competitivo.

Além da resistência à gestão, um projeto *Big Data* pode falhar por outros motivos, como por exemplo: falta de capacitação da equipe, aquisições de tecnologia sem conhecimento, obtenção de dados desnecessários e confiar mais na intuição do que nos dados.

Portanto, de acordo com Pereira (2016) como o *Big Data* ainda é recente, ou seja, apenas nos últimos anos é que as empresas se depararam com a necessidade de fazer um melhor uso de seus dados internos e utilizar os dados externos para melhores análises e tomadas de decisão, muitas pesquisas ainda são necessárias para auxiliar a implantação de *Big Data* nas corporações. Por conta disso, definiu-se o objetivo principal deste trabalho, que é apresentar uma proposta de aprendizagem de conhecimento de processo de projetos *Big Data* com o *framework* de gestão Scrum.

1.2 Objetivo

O objetivo geral do trabalho é apresentar como a metodologia ágil Scrum pode ser aplicada para aprendizado de projetos *Big Data* a fim de melhorar problemas de engenharia e *software* e aprender um *framework* de gestão.

1.3 Método de trabalho

O estudo sobre a metodologia ágil Scrum e *Big Data* será realizado por meio de pesquisa bibliográfica. Esta atividade colabora para entender como ocorre a gestão e o processo de projetos que utilizam a metodologia Scrum, o processo de projetos *Big Data* e sua importância no negócio.

Posteriormente, será realizada uma análise para identificar quais artefatos e eventos da metodologia Scrum poderão ser utilizados para que se aprenda as etapas de projetos *Big Data*.

Uma vez que a análise foi definida, será explicado como o processo de aprendizagem para projetos *Big Data* ocorrerá utilizando a metodologia Scrum, ou seja, será detalhado o processo de aprendizagem do *Big Data* através das fases do Scrum, da análise e desenvolvimento a serem realizados até a sua implementação.

E, finalizando a sequência do trabalho, será explicado um exemplo de utilização do novo processo e será realizada a avaliação dos benefícios a serem obtidos com a utilização da metodologia.

1.4 Organização do texto

Este trabalho está estruturado da seguinte forma: o primeiro capítulo apresenta uma introdução ao assunto, descrevendo o problema de pesquisa, o objetivo e a justificativa. O capítulo dois apresentará as principais etapas de desenvolvimento de um projeto *Big Data*. O capítulo três tem como finalidade

apresentar o estudo do *framework* Scrum, seus pontos fracos e sua definição, papéis, artefatos e eventos que o compõe. O capítulo quatro é focado no desenvolvimento da pesquisa realizada e análise da solução proposta, ou seja, explicação do processo de Sprints de aprendizagem. Por fim, o capítulo cinco apresenta as conclusões e os trabalhos futuros.

2 *Big Data* e dificuldades em projetos de TI

O objetivo deste capítulo é de apresentar aspectos da tecnologia *Big Data* que introduzem dificuldades técnicas na concepção de soluções de software. A primária dificuldade revela-se no tempo de aquisição de conhecimento tecnológico na janela de execução do projeto. O conteúdo presente neste capítulo visa contribuir com o entendimento de projetos *Big Data*, desde o seu conceito até seu processo. As dificuldades de condução de um projeto envolvendo a tecnologia de *Big Data* serviram como ponto de inspiração para a realização desta monografia.

Neste capítulo apresenta-se o estudo conceitual do *Big Data*. Na seção 2.1 define-se *Big Data*, incluindo sua história, evolução e importância no mundo atual. Em seguida, na seção 2.2, descrevem-se as principais etapas em um hipotético projeto baseado em *Big Data*.

2.1 Definição de *Big Data*

O marco da primeira era da informação foi em 1954 com a implementação de um sistema de folha de pagamento pela General Electric (GE). Com a criação da Internet, ocorreu a segunda era da informação, que ficou chamada de “a era da Rede”, que junto com a globalização tornou o ambiente empresarial ainda mais complexo (MINELI, CHAMBERS E DHIRAJ, 2013 apud NESELLO e FACHINELLI, 2014). De acordo com Rapporteur (2010, apud NESELLO e FACHINELLI, 2014), a disseminação das redes sociais, a criação da computação em nuvem e as novas tecnologias que surgiram originaram um número muito grande de informações, gerando assim, a terceira era da informação, que se refere ao *Big Data*, ou seja, a era do Dado.

Pode-se resumir que *Big Data* é o termo utilizado para definir a enorme quantidade de dados que são gerados todos os dias e podem conceder vantagens e inovações nas competitividades organizacionais. (TAURION, 2013 apud DISNER, 2014).

Há basicamente dois tipos de dados: os dados estruturados e os dados não-estruturados. Os dados estruturados são os dados armazenados em tabelas de banco de dados relacionais. E os dados não estruturados são, em sua maioria, compartilhados por pessoas e capturados das mídias sociais. Estes dados podem ser textuais (por exemplo, e-mails, conteúdo de blogs, postagens de mídias sociais, mensagens de texto, buscas da internet, entre outros) e não-textuais (por exemplo, vídeos, imagens, áudios, etc). (TAURION, 2013a apud PEREIRA, 2015 e EREVELLES et al., 2015)

Os dados estão sendo gerados constantemente e de forma rápida e se não forem analisados, muita informação será perdida. A diversidade destes dados está transformando a tomada de decisões de *marketing*. Já que, uma boa decisão é a combinação de dados e informações relevantes, e fornecer um grande volume de dados variados em um pequeno espaço de tempo, é o que o *Big Data* faz.

Essas três dimensões auxiliam na definição do *Big Data*, comumente referido como os três Vs: volume, velocidade e variedade. (IBM, 2012; Lycett, 2013; Oracle, 2012 apud EREVELLES et al., 2015).

- Volume: apesar de abstrato, o volume é o maior destaque e a característica mais corrente atribuída ao *Big Data*. (PEREIRA, 2015)

De acordo com Erevelles et al. (2015), atualmente, o volume de dados não-estruturados é medido em petabytes, exabytes ou zettabytes. Para que as empresas consigam enfrentar este crescente volume de dados, estima-se que o mercado global de *software*, *hardware* e serviços dobre de tamanho a cada dois anos para conseguir armazenar e analisar os dados não-estruturados. (IDC, 2014 apud EREVELLES et al., 2015).

- Velocidade: na maioria das vezes, para obter um resultado eficaz, analisar dados quase em tempo real é fundamental, pois um dado coletado ontem, já estará ultrapassado hoje. Logo, a velocidade está associada à agilidade de obtenção e análise dos dados. (CABRAL; SAID, 2014 apud PEREIRA, 2015 e DISNER, 2014)

Em oposição as análises dos dados tradicionais, ou seja, estruturados, para o *Big Data* é necessário obedecer um fluxo contínuo de coleta e análise dos dados não-estruturados. (DAVENPORT, 2014, p.16 apud PEREIRA, 2015).

- Variedade: a variedade refere-se às diferentes fontes do *Big Data* que fornecem uma rica diversidade dos dados, diferentemente dos dados

tradicionais utilizados no passado. (PEREIRA, 2015 e EREVELLES et al., 2015)

A grande diferença entre os dados utilizados no passado e atualmente, no *Big Data*, é a alteração de dados transacionais estruturados para os dados não-estruturados que são gerados por e-mails, mídias sociais (Facebook, Twitter, YouTube e outros), documentos eletrônicos, mensagens instantâneas, sensores, RFID, fotos, vídeos, etc. (Integreon Insight, 2012 apud EREVELLES et al., 2015).

Os três Vs descritos acima são utilizados para definir e diferenciar o *Big Data* de conjuntos de dados em larga escala, porém, ainda há os Vs que são importantes no processo de coleta, análise e extração de informações de dados *Big Data*: veracidade e valor (Ebner, Bühnen e Urbach, 2014; Lycett, 2013 apud EREVELLES et al., 2015).

- Veracidade: para que as tomadas de decisões derivadas das análises de dados sejam assertivas, é essencial que os dados utilizados sejam verdadeiros. (DISNER, 2014)

Isto é, a veracidade enfatiza a necessidade de conhecimento da qualidade dos dados, pois nem todos os dados obtidos são precisos. Portanto, a veracidade do *Big Data* está sendo um importante problema atualmente, visto que o volume, a velocidade e a variedade de dados estão aumentando constantemente (IBM, 2012; Oracle, 2012 apud EREVELLES et al., 2015). Porém, com análises e estatísticas de amplos volumes de dados é possível compensar informações incorretas.

- Valor: é de extrema importância que ao se implementar um projeto *Big Data* em uma organização que ele obtenha retorno positivo do investimento realizado. (TAURION, 2013 apud DISNER, 2014)

2.2 Projeto *Big Data*

Para iniciar um projeto com foco no processamento de uma enorme quantidade de dados não estruturados, um projeto *Big Data*, precisa-se entender o contexto do negócio e os problemas que serão atacados, ou seja, definir o escopo

do problema utilizando os 5Vs que foram apresentados previamente e o valor que ele gerará à empresa.

Há uma enorme dificuldade para a realização desta atividade, uma vez que o tempo necessário de assimilação do problema não pode ser muito alto para não exceder o cronograma do projeto, e, também, se a equipe apenas possuir conhecimento teórico, o escopo definido do problema não necessariamente será o melhor. Por outro lado, se nem conhecimento teórico a equipe possuir, dificilmente o escopo definido estará correto, o que acarretará em problemas futuros, ou seja, na entrega do projeto.

Com o problema já definido, deve-se identificar quais dados serão necessários para o projeto. Analisando os dados existentes para o *Big Data*, é importante salientar que existem duas fontes de dados: os dados internos e os externos. Os dados internos são dados estruturados, não-estruturados ou semiestruturados que são provenientes da organização. Os dados externos são os dados não-estruturados provenientes da Internet que estão disponíveis em mídias sociais, informações de GPS, vídeos, fotos, informações de produto concedidas gratuitamente pelo fornecedor, reclamações de clientes postados em sites de regulamentação, entre outros. (SATHI, 2012 apud NESELLO e FACHINELLI, 2014)

Os dados não-estruturados, na maioria das vezes, são os dados que as empresas utilizarão para análise a fim de realizar projetos *Big Data*. (PEREIRA, 2015)

Outra etapa importante é definir qual base de dados será utilizada, de acordo com o dado a ser analisado. A modelagem de dados tem a finalidade de representar uma faceta da realidade no computador, viabilizando o seu uso de acordo com algum propósito analítico. Ela é realizada em várias etapas ao longo de um processo de desenvolvimento de *software*, pois a natureza do dado a ser utilizado variará de acordo com regras de negócio e necessidades do sistema. (BARBIERI, 1994 apud ALVAREZ, 2013)

Por este motivo, o armazenamento de dados pode ser realizado de diferentes formas, pois alguns dados são estruturados e, por este motivo, armazenados em uma base de dados relacional. Enquanto outros dados, como os não estruturados, compostos de registros de Internet, documentos, fotos e vídeos, fonte de informações produzidas pelo ser humano, como: dados de mídias sociais e os

gerados a partir de interações com uma página na web são armazenados em bases de dados não-relacionais. (HURWITZ et al, 2013, apud ALVAREZ, 2013).

Após armazenar os dados, a próxima etapa será definir quais algoritmos e tecnologias serão necessários para analisar estes dados. A maioria dos dados do *Big Data* precisam ser analisados para que seja possível criar tomadas de decisões mais eficientes e que apresentem um resultado mais rápido e assertivo. Devido a essas características, a extração e o processamento de *Big Data* necessita de ferramentas e técnicas especiais, como o *Data Mining* (mineração de dados) (DISNER, 2014).

O *Data Mining* faz parte do Business Intelligence (BI) e indica o processo de extração e análise de um conjunto de dados para identificar qualquer regularidade, extraindo novos conhecimentos e regras de aplicações significativas. Seu principal objetivo é "extrair informações" úteis de um banco de dados e transformá-las em um padrão, que será utilizado para uso posterior. (GIACALON e SCIPPACERCOL, 2016)

Nesta etapa de processo, há também uma grande dificuldade técnica, pois se os desenvolvedores não conhecerem a tecnologia que estão utilizando, eles terão que aprender durante o andamento do projeto, antes do seu término. Por este motivo, é necessário que os desenvolvedores possuam experiência neste tipo de tecnologia para se gerencie tal risco do projeto.

3 Scrum

O objetivo deste capítulo é apresentar o *framework* Scrum, suas definições, eventos e artefatos que os compõe. O Scrum procura ajudar nas dificuldades encontradas em metodologias tradicionais, como por exemplo: alterações de escopo após as Sprints, a análise de risco dos itens a serem desenvolvidos em cada Sprint, entregas em intervalos regulares. Porém, como não há um processo fixo para definição da arquitetura, por exemplo, a Equipe de Desenvolvimento deve ser formada por pessoas com experiência e conhecimento que consigam desenvolver sistemas de forma a cumprir prazos e ser efetiva.

O Scrum é baseado no empirismo e um dos seus pilares é a adaptação, que, neste contexto, é sobre melhorias contínuas, ou seja, a habilidade de se adaptar de acordo com os resultados observados. O Scrum traz bons resultados a quem o utiliza porque adere aos princípios ágeis subjacentes de entrega incremental iterativa, coletando frequentemente os comentários dos clientes e abraçando mudanças.

3.1 Métodos ágeis e solução de problemas

As metodologias ágeis nasceram com o objetivo de ampliar o foco nas pessoas e não em processos de desenvolvimento. Elas também propuseram elementos para administrar situações relacionadas a desperdício de tempo decorrentes da escrita de documentos. Conseqüentemente abriram a possibilidade de disponibilizar mais o tempo com a resolução de problemas de forma iterativa. (SOARES, 2004)

O Scrum é uma metodologia ágil para gestão e planejamento de projetos de *software* desenvolvida por Jeff Sutherland e sua equipe no início da década de 90. Ele é baseado nos princípios do Manifesto para o Desenvolvimento Ágil de *Software*. (BECK, et al., 2001)

Para Schwaber e Sutherland (2016), Scrum é um *framework*¹, que se pode empregar diversos processos ou técnicas, de maneira que quem o utiliza pode melhorá-las. Portanto, ele permite que as pessoas resolvam problemas complexos e adaptativos, enquanto provêm produtos criativos com maior valor possível.

De acordo com Sabbagh (2013, p.17), “Scrum é um *framework* ágil, simples e leve, utilizado para a gestão do desenvolvimento de produtos complexos imersos em ambientes complexos. Scrum é embasado no empirismo e utiliza uma abordagem iterativa e incremental para entregar valor com frequência e, assim, reduzir os riscos do projeto. ”

3.2 Deficiências do papel Development Team (Time de Desenvolvimento)

O Time Scrum é composto por: Scrum Master, Product Owner (Dono do Produto) e Development Team (Time de Desenvolvimento). Eles são multifuncionais e auto organizáveis, ou seja, eles possuem todas as competências essenciais para finalizar um trabalho e escolhem a melhor forma de executá-lo. Não há ninguém fora da equipe que os auxilie. (SCHWABER e SUTHERLAND, 2016)

3.2.1 Product Owner (Dono do Produto)

O Product Owner (ou dono do produto), também chamado de P.O., é um papel exercido por apenas uma pessoa em um Time de Scrum. Ele é o responsável por garantir e maximizar o retorno sobre o investimento no produto para os clientes do projeto, através do trabalho do Time de Desenvolvimento. Ou seja, em um projeto que utiliza Scrum, ele é o gerente do produto. (SABBAGH, 2013)

“O Product Owner define o produto e toma as decisões de negócios relativas a seu desenvolvimento a partir das necessidades dos clientes do projeto e demais

¹ Conjunto de conceitos utilizados na resolução de um problema de caráter específico, isto é, ele pode se ajustar a diversos contextos de gerenciamento de projetos com um processo simples, que pode ser complementado.

partes interessadas, alinhado com ou em direção aos objetivos da organização.” (SCHWABER e SUTHERLAND, 2016)

Conforme descrito acima, uma das atividades do P.O. é realizar levantamento das necessidades do cliente e inseri-las em uma lista chamada, Product Backlog. Ele é o único que pode adicionar, remover ou alterar itens desta lista conforme solicitação do cliente. Uma outra grande atividade do Product Owner é realizar a gestão de releases do produto, isto é, cabe a ele definir a melhor estratégia de planejamento e a modificar quando necessário. Esta atividade é realizada com base no planejamento do Time de Desenvolvimento para que os riscos do projeto sejam reduzidos. (SABBAGH, 2013, pg. 82-84)

3.2.2 Scrum Master

A responsabilidade do Scrum Master é garantir o entendimento e aplicação do Scrum, com o objetivo que o Time Scrum entenda a teoria, práticas e regras do Scrum. (SCHWABER e SUTHERLAND, 2016)

“Ou seja, utilizando-se de seu conhecimento de Scrum, habilidade de lidar com pessoas, técnicas de facilitação e outras técnicas, o Scrum Master ajuda o Product Owner e Time de Desenvolvimento a serem mais eficientes na realização do seu trabalho.” (SABBAGH, 2013, pg. 91)

3.2.3 Development Team (Time de Desenvolvimento)

O Time de Desenvolvimento é estruturado e liberado pela organização para gerenciar o desenvolvimento do produto. É sua responsabilidade determinar como o produto será desenvolvido tecnicamente e, por isso, é responsável e responsabilizado por seus resultados. O Time de Desenvolvimento possui as seguintes características: multifuncional, auto-organizado, pequeno (formado de 3 a 9 pessoas) e focado em metas pré-estabelecidas com o Product Owner. (SCHWABER e SUTHERLAND, 2016; SABBAGH, 2013)

Quando o Product Backlog é apresentado ao Time de Desenvolvimento pelo Product Owner, é de responsabilidade do Time mostrar o que é tecnicamente possível de execução, o custo de cada um dos itens e se haverá algum trabalho técnico suplementar, e se houver, ele deverá ser adicionado ao Product Backlog. (SABBAGH, 2013, pg.82)

3.3 Engrenagens que movimentam o Scrum

Os artefatos do Scrum são definidos para aumentar a transparência das informações, de maneira que todos os envolvidos consigam ter a mesma visão do negócio. (SCHWABER e SUTHERLAND, 2016)

Há apenas quatro artefatos no Scrum: o Product Backlog, o Sprint Backlog, o Incremento do Produto e a Definição de “Pronto”.

Porém, há duas Metas de Negócios que são muito importantes: Visão do Produto e Meta do Sprint. A Visão do Produto é o objetivo a ser atingido através do desenvolvimento do produto. Ela é definida antes do início dos Sprints e vive ao longo do desenvolvimento. Por outro lado, a Meta do Sprint é obrigatoriamente definida durante a reunião de Sprint Planning de cada Sprint do projeto. Cada funcionalidade criada pelo Time de Desenvolvimento é um passo ao atingimento da Meta do Sprint. (SABBAGH, 2013, pg. 160)

3.3.1 Product Backlog

O Product Backlog é uma lista de necessidades ou objetivos de negócio do cliente e/ou demais partes interessadas, escrita em linguagem do negócio (onde todos devem possuir o mesmo entendimento). Ele contém tudo o que se espera que seja desenvolvido pelo Time de Desenvolvimento, desde melhorias solicitadas até correções de defeitos. (SABBAGH, 2013, pg. 111-112)

O Product Backlog é dinâmico e está sempre em aperfeiçoamento com itens sendo adicionados, removidos, reordenados ou modificados de acordo com a

necessidade do produto. Pois enquanto o produto existir, ele também existirá e nunca estará completo. (SCHWABER e SUTHERLAND, 2016)

Os itens do topo da lista do Product Backlog devem possuir um maior detalhamento de descrição e estimativas de tempo de desenvolvimento mais precisas do que os itens de ordem mais baixas, pois são prioritários e serão desenvolvidos primeiro pelo Time de Desenvolvimento. Enquanto os itens menos prioritários são de alta granularidade e com estimativas com certo grau de incertezas. (SCHWABER e SUTHERLAND, 2016)

Não há padrão definido para o Product Backlog no Scrum. Apenas é essencial que ele tenha itens em sequência, prioridades diferentes e seja reordenável com facilidade. (SABBAGH, 2013, pg. 112)

3.3.2 Sprint Backlog

No Sprint Planning Meeting, os itens do Product Backlog são priorizados pelo Product Owner e apresentados ao Time de Desenvolvimento. O Time de Desenvolvimento solicita ao Product Owner detalhes indispensáveis para que ele seja capaz de selecionar uma quantidade de itens do Product Backlog, que acredite que complete sua capacidade de afazeres em um Sprint. Assim sendo, o Time de Desenvolvimento define quais itens conseguirá completar no decorrer do Sprint que se iniciará. Os itens selecionados são movidos do Product Backlog para o Sprint Backlog. (SABBAGH, 2013)

O Sprint Backlog pertence exclusivamente ao Time de Desenvolvimento e retrata uma imagem em tempo real do trabalho a ser executado. Ele também possui um bom nível de detalhamento para que seja possível acompanhar seu progresso durante a Reunião Diária. No decorrer do Sprint, o Sprint Backlog é atualizado para refletir o trabalho restante para que a Meta do Sprint seja alcançada, isto é, pode haver alterações no plano, como adição ou remoção de novas tarefas e reestimativas. Porém, uma grande quantidade de mudanças pode dificultar ou impossibilitar o atingimento da Meta do Sprint, o que significa, geralmente, que o Sprint foi mal executado. (SCHWABER e SUTHERLAND, 2016)

3.3.3 Incremento do Produto

Na visão de Sabbagh (2013), o Time de Desenvolvimento desenvolve os itens selecionados no Sprint Backlog durante o Sprint, com o objetivo de atingir a Meta do Sprint. O Incremento do Produto é o resultado deste trabalho que é o conjunto de todos os itens concluídos do Product Backlog até o final de um Sprint. Logo, este incremento possui novas funcionalidades e melhorias dos itens desenvolvidos previamente. (SCHWABER e SUTHERLAND, 2016)

No término do Sprint o incremento novo necessita estar “Pronto”, isto é, estar usável e atender a definição de “Pronto” do Time Scrum. Independente do Product Owner liberá-lo aos usuários ou não, ele deve estar na condição utilizável. (SCHWABER e SUTHERLAND, 2016)

3.4 Eventos

O Scrum possui alguns eventos que são aplicados para criação de rotina e também diminuição da ocorrência de reuniões não planejadas. Todos estes eventos são time-boxed, isto é, possuem um tempo pré-determinado, o que faz com que o evento seja realizado no tempo estabelecido, evitando desperdícios. (SCHWABER e SUTHERLAND, 2016)

Para Sabbagh (2013), os eventos do Scrum são chamados de: Sprint, e as reuniões realizadas durante o ciclo, que são: Sprint Planning (Planejamento do Sprint), Daily Scrum (Reunião Diária), Sprint Review (Revisão do Sprint) e Sprint Retrospective (Retrospectiva do Sprint).

O Sprint possui uma característica diferente dos outros eventos, uma vez iniciada, sua duração não pode ser aumentada ou diminuída. Por outro lado, os outros eventos podem ser concluídos sempre que o objetivo for atingido. (SCHWABER e SUTHERLAND, 2016)

Os eventos são nomeadamente projetados para possuir transparência e inspeção criteriosa. Não incluir algum dos eventos no Scrum acarretará na redução

da transparência e na perda de oportunidades para inspeção e adaptação. (SCHWABER e SUTHERLAND, 2016)

3.4.1 Sprint

O Sprint pode ser definido como um miniprojeto, onde devem ser desenvolvidos itens determinados pela sua importância, cujo foco é atingir a Meta do Sprint e gerar o Incremento do Produto a partir do Sprint Backlog. (SABBAGH, 2013, pg. 195-196)

Durante o Sprint, acontecem, além do desenvolvimento, as reuniões diárias (Daily Scrum), a reunião de Revisão do Sprint (Sprint Review), a reunião de Retrospectiva do Sprint (Sprint Retrospective) e outras atividades que acontecem com a participação do Time de Desenvolvimento. Todos esses eventos devem ocorrer dentro do timebox do Sprint. (SABBAGH, 2013, pg. 196)

O projeto que utiliza o *framework* Scrum funciona por completo dentro de Sprints. Eles ocorrem um após o outro, sem pausas ou intervalos. Logo, não se paralisa um Sprint após sua finalização ou a interrompe temporariamente para sanar questões sobre o projeto. Todos os acontecimentos do projeto com Scrum é conduzido para dentro do Sprint. (SABBAGH, 2013, pg. 196)

3.4.2 Sprint Planning (Planejamento do Sprint)

O Sprint Planning ou planejamento do Sprint é a organização do ciclo de desenvolvimento do Sprint. Ele se inicia no primeiro dia do Sprint em seu primeiro instante. (SABBAGH, 2013, pg. 203)

A reunião de Sprint Planning é obrigatoriamente composta pelo Product Owner, Time de Desenvolvimento e Scrum Master, que atua como facilitador. Nesta reunião o Product Owner e o Time de Desenvolvimento negociam o que será desenvolvido, a Meta do Sprint e como os itens do Sprint Backlog serão desenvolvidos. (SABBAGH, 2013, pg. 204)

A reunião de Sprint Planning possui um time-box de no máximo oito horas para uma Sprint de um mês de duração. Para Sprints menores, este o time-box é menor. O papel do Scrum Master é assegurar que o evento aconteça e que os participantes percebam qual seu propósito. (SCHWABER e SUTHERLAND, 2016)

Na reunião de planejamento da Sprint é negociada a Meta do Sprint e seu Sprint Backlog. Com isso, o Time de Desenvolvimento gera o melhor planejamento de acordo com o conhecimento que possuem. Porém, no decorrer do Sprint, o plano do Sprint Backlog irá evoluir, conforme o Time de Desenvolvimento adquire maior conhecimento ao executá-lo. (SABBAGH, 2013, pg. 204-206)

3.4.3 Daily Scrum (Reunião Diária)

No Scrum, a Reunião Diária acontece normalmente no mesmo local e hora, é um evento curto, cujo time-boxed é de quinze minutos, realizada diariamente pelo Time de Desenvolvimento para inspecionar o trabalho que foi realizado desde a última Reunião Diária, alinhar as atividades e criar um planejamento para as próximas 24 horas. (SCHWABER e SUTHERLAND, 2016)

De acordo com Sabbagh (2013), há um padrão utilizado para dirigir as questões na reunião diária, cada integrante da equipe deve falar o que fez desde a última reunião, o que pretende fazer até a próxima reunião e quais são os obstáculos que impedem a realização do trabalho.

3.4.4 Sprint Review (Revisão do Sprint)

A reunião de Sprint Review (ou revisão do Sprint) tem a finalidade de demonstrar o trabalho pronto, isto é, o desenvolvimento de todos os itens do Sprint Backlog, realizado pelo Time de Desenvolvimento e Product Owner durante a Sprint aos envolvidos do projeto. O principal objetivo é obter o feedback dos envolvidos sobre o Incremento do Produto feito, que servirá como base ao Product Owner para atualizar o Product Backlog para os próximos Sprints. (SABBAGH, 2013, pg. 220)

“A reunião de Sprint Review acontece no último dia do Sprint, antes da reunião de Sprint Retrospective, e tem a duração máxima de quatro horas para Sprints de um mês ou proporcionalmente menos para Sprints mais curtas. “ (SABBAGH, 2013, pg. 220)

O efeito da Reunião de Revisão da Sprint é um Backlog do Produto corrigido, que determinará o possível Backlog do Produto para as Sprints seguintes. O Backlog do Produto também pode ser atualizado para atender novas funcionalidades. (SCHWABER e SUTHERLAND, 2016)

3.4.5 Sprint Retrospective (Retrospectiva do Sprint)

“A Retrospectiva da Sprint é uma oportunidade para o Time Scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima Sprint.” (SCHWABER e SUTHERLAND, 2016)

Durante a reunião de Sprint Retrospective, o Time de Desenvolvimento e Product Owner analisam os processos de trabalho, as dinâmicas, os comportamentos, as ferramentas utilizadas e o ambiente da Sprint que está terminando para poder planejar as melhorias importantes, ou seja, eles identificam o que ocorreu bem no Sprint para ser mantido no próximo Sprint e o que deve ser melhorado, identificando as causas raízes das dificuldades encontradas e realizar planos de ação com maneiras práticas e eficientes de melhorar o processo. (SABBAGH, 2013, pg. 226)

A reunião de Sprint Retrospective é uma cooperação e nunca deve ser utilizada para realizar acusações ou discussões. Seu foco é realizar a melhoria do processo de trabalho do Time de Scrum e não apontar erros individuais, isto é, seu objetivo é aprender com os erros cometidos. (SABBAGH, 2013, pg. 226-234)

4 Scrum e Sprints para aprendizagem de projetos *Big Data*

Neste trabalho o *framework* Scrum será adaptado para acomodar Sprints de natureza não convencional. O presente estudo baseia-se em uma pesquisa de caráter descritivo, qualitativo, longitudinal e não exploratório. A pesquisa identificou problemas encontrados nos projetos de *Big Data* e limitou-se nestas amostras: falta de conhecimento de *Big Data* dos engenheiros de *software*, aquisições de tecnologia sem conhecimento de utilização, não entender o problema e não saber quais dados serão necessários para tal tratativa. Através deste método de pesquisa, espera-se entender como as fases do projeto *Big Data* podem se fundir com os artefatos e eventos do Scrum, para então, propor uma solução experimental de aprendizagem de *Big Data*.

A seguir será apresentado o processo alcançado através de levantamento dos conceitos de Scrum e do *Big Data*, suas compatibilidades de processos, assim como as definições dos papéis do Scrum em um cenário de projeto de aprendizagem de *Big Data* e ainda as aplicações dos eventos e artefatos neste mesmo projeto.

4.1 Processo

Utilizar Scrum em projetos de *Big Data* pode corrigir alguns problemas comuns que ocorrem em projetos deste tipo, entre eles: iniciar o projeto com uma enorme quantidade de dados, não possuir o problema bem definido, não saber quais dados serão necessários, falta de suporte do usuário e falta de clareza de definição de papéis.

Porém, apesar do *framework* de gestão Scrum parecer ser a melhor escolha para se gerenciar um projeto de *Big Data*, porque ele acomodará alterações durante o desenvolvimento do projeto de uma maneira melhor do que uma metodologia cascata, por exemplo. Por si só ele é insuficiente para garantir o sucesso do projeto, pois o problema de domínio de engenharia de *software* por parte do Time de Desenvolvimento continuará existindo.

Por este motivo, a implantação do *framework* Scrum nos projetos de *Big Data* aconteceria com a utilização de um projeto piloto de aprendizagem que conteria todos os eventos e artefatos do Scrum, as principais fases de um projeto *Big Data*, isto é, utilizando um Product Backlog de etapas técnicas, cujos itens seriam as etapas do desenvolvimento do projeto.

No capítulo 2 foi possível aprender e analisar o processo Scrum. Na Tabela 1 encontra-se um resumo de fases, processos e papéis definidos em cada etapa do Scrum.

Fase	Processos	Papéis
Inicial	1. Criar Visão do Projeto	Product Owner
	2. Criar Priorização do Product Backlog	Product Owner
Planejamento e Estimativas	3. Criar User Stories	Product Owner
	4. Aprovar, Estimar e Fechar User Stories	Development Team, Product Owner
	5. Criar Tarefas	Development Team
	6. Estimar Tarefas	Development Team
	7. Criar Sprint Backlog	Development Team
Implementação	8. Criar entregáveis	Development Team
	9. Conduzir Standup Diário	Scrum Master
	10. Manter Priorização do Product Backlog	Product Owner
Revisão e Retrospectiva	11. Demonstrar e Validar o Sprint	Development Team, Product Owner
	12. Realizar Revisão do Sprint	Development Team, Product Owner e Scrum Master
	13. Realizar Retrospectiva do Sprint	Development Team, Product Owner e Scrum Master
Entregas	14. Entregar Produto	Development Team
	15. Retrospectiva do Projeto	Development Team, Product Owner e Scrum Master

Tabela 1 – Resumo Scrum

Fonte: Criado pelo autor

No capítulo 3, foram descritos os principais conhecimentos que um analista *Big Data* precisa saber para realizar projetos de sucesso. Com isso, a Tabela 2 resume as etapas de um projeto *Big Data*.

Fase	Processos	Papéis
Definição do Problema	1. Entender contexto do negócio	Analista <i>Big Data</i> , Stakeholder
	2. Entender o problema	Analista <i>Big Data</i>
Análise	3. Definir dados	Analista <i>Big Data</i>
	4. Verificar se os dados estão disponíveis	Analista <i>Big Data</i>
	5. Verificar se os dados já existem na empresa	Analista <i>Big Data</i>

	6. Validar se não há risco judicial na utilização dos dados	Analista <i>Big Data</i> , Advogado
Desenvolvimento	7. Definir algoritmos e tecnologias	Analista <i>Big Data</i>
	8. Realizar correlação dos dados do problema	Analista <i>Big Data</i>
Implantação	9. Geração do dado correto	Analista <i>Big Data</i>
	10. Algoritmo escolhido funcionando	Analista <i>Big Data</i>
	11. Ação com base nos resultados obtidos	Analista <i>Big Data</i> , Stakeholder

Tabela 2 – Resumo Projeto *Big Data*

Fonte: Criado pelo autor

E a junção de ambos processos resultariam no processo descrito abaixo:

4.1.1 Definição de Papéis

De acordo com a definição de Scrum, há 3 papéis que compõem o Time Scrum: Scrum Master, Product Owner e Time de Desenvolvimento. Por outro lado, no desenvolvimento de um projeto *Big Data*, temos: Analista *Big Data*, Advogado e Stakeholder.

Para este projeto de aprendizagem, os papéis acima serão definidos da seguinte maneira:

- Scrum Master – Por ser um papel específico do *framework* Scrum, ele será definido por um próprio Scrum Master, ou seja, garantirá que a equipe acate e siga os valores e as práticas do Scrum;
- Product Owner – Especialista da equipe de desenvolvimento que será responsável por manter o Product Backlog e repassar o conhecimento técnico para a equipe que não o possui;
- Time de Desenvolvimento – Equipe de desenvolvimento de projeto *Big Data* (analista funcional, programador, analista de testes, arquiteto de *software*) que não é formada apenas por especialistas;
- Analista *Big Data* – Será definido como Time de Desenvolvimento;
- Advogado – Neste projeto de aprendizagem ele não existirá, visto que não haverá nenhum desenvolvimento a ser implantado em ambiente produtivo;

- Stakeholder – Neste projeto de aprendizagem ele não existirá, visto que ele não solicitará nenhuma nova funcionalidade a ser entregue.

Portanto, neste projeto os papéis existentes serão: Scrum Master, Product Owner e Time de Desenvolvimento, isto é, o Time Scrum.

4.1.2 Artefatos e Eventos

Os artefatos e eventos existentes atualmente no *framework* Scrum serão mantidos, e, alguns novos artefatos serão adicionados ao processo.

4.1.2.1 O Product Backlog

O Product Backlog será a lista detalhada dos itens que precisam ser aprendidos e/ou aprimorados tecnicamente para transformar projetos *Big Data* em sucesso.

Logo, o Product Owner escreverá o Product Backlog com as atividades realizadas em cada etapa de um projeto *Big Data* (entender o problema, definir os dados, preparar e modelar os dados utilizando algoritmos e tecnologias, correlacionar de dados e implantar o projeto) que ele considera que a equipe esteja tendo mais dificuldade durante o desenvolvimento de uma nova funcionalidade.

O Product Backlog não precisará estar completo logo no começo do projeto, ele poderá ser incrementado no decorrer do tempo, visto que novas dúvidas e tecnologias irão surgir.

4.1.2.2 Sprints educacionais

Após a criação do Product Backlog pelo P.O., as Sprints devem ser definidas. Neste projeto piloto de aprendizagem, a Sprint educacional terá duração de 2 semanas, para que seja possível que o Time de Desenvolvimento aprenda e consiga

tirar possíveis dúvidas com o P.O., neste caso, o especialista da equipe. E também, como a Sprint de aprendizagem pode acontecer entre as Sprints de um projeto em andamento, ela não prejudicaria o desenvolvimento do projeto.

A primeira Sprint educacional será criada no Sprint Planning (Planejamento do Sprint) onde serão realizadas as estimativas de cada item do Product Backlog, a criação de user stories (se necessário) e a criação do Sprint Backlog. O Product Owner priorizará os itens do Product Backlog que ele identifica que a equipe está tendo mais dificuldades e os descreverá à equipe. A equipe auxiliará na seleção dos itens a serem aprendidos durante a Sprint que está por começar.

Os itens selecionados serão transferidos do Product Backlog para o Sprint Backlog. Neste momento, o Time de Desenvolvimento criará as tarefas relacionadas ao Sprint Backlog de acordo com cada item do Product Backlog com o objetivo de aprender/aprimorar o conhecimento desta etapa.

4.1.2.3 Dicionário de conhecimento

Um dos novos artefatos será o Dicionário de Conhecimento que consistirá em uma lista organizada de todos os elementos aprendidos com definições precisas, para que um novo membro na equipe consiga adquirir conhecimento já repassado sem a necessidade de uma nova Sprint.

A construção do Dicionário de Conhecimento será uma das tarefas mais trabalhosas e consumirá um grande tempo do Time de Desenvolvimento durante a Sprint. Por outro lado, será um dos entregáveis mais importantes, pois servirá de consulta aos analistas da equipe e também como guia de aprendizagem aos novos analistas que farão parte da equipe.

No início da Sprint, o Time de Desenvolvimento deverá preencher algumas informações no Dicionário de Conhecimento sobre os itens que serão tratados no Sprint Backlog:

- Sprint;
- Participantes da Sprint;
- Descrição dos itens do Backlog da Sprint;

Vide exemplo abaixo:

Sprint:

Participantes:

Descrição dos itens do Backlog a serem aprendidos:

Figura 1 – Exemplo de Descrição da Sprint de Aprendizagem no Dicionário de Conhecimento

Fonte: Criado pelo autor

No decorrer da Sprint, será necessário adicionar informações aos itens do Dicionário de Conhecimento com o aprendizado obtido. Os itens do Dicionário de Conhecimento serão compostos pelas informações abaixo:

- Item do Backlog;
- Sua definição;
- Condições de utilização;
 - Apropriada quando:
 - Expressar procedimentos;
 - Existência de diferentes utilizações;
 - Regras utilizadas;
- Materiais utilizados - ata de reunião, anotações realizadas por alguma palestra, treinamento virtual, ou algum outro material relevante;
- Links de sites;
- Fluxograma, se necessário;
- Árvore de decisão, se necessário;
- Exemplos concedidos que agregam muito valor ao aprendizado;
- Evidências de sucesso de um projeto realizado, após a finalização da Sprint do projeto;

Ao final de cada Sprint, o Time de Desenvolvimento deve ter finalizado a inserção de informações do Dicionário de Conhecimento dos itens do Sprint Backlog e o Scrum Master deve garantir que o Dicionário de Conhecimento está atualizado.

Ele deve ser atualizado apenas no final da Sprint, pois nem todo o conteúdo pesquisado durante a Sprint auxiliou no entendimento do processo como um todo e apenas os itens que foram relevantes devem ser adicionados.

O Dicionário de Conhecimento deverá ser realizado em *software* local ou em nuvem, e poderá ser feito em planilha, documento virtual, *software* de projetos Scrum, ou outro *software*.

Abaixo encontra-se exemplo do Dicionário de Conhecimento:

Item do Backlog:

Definição do item:

Condição de utilização:

Materiais utilizados na Sprint:

Links úteis:

Fluxos

Exemplos úteis:

Evidências de sucesso de projeto após Sprint:

Figura 2 – Exemplo de Dicionário de Conhecimento - Item do Backlog

Fonte: Criado pelo autor

4.1.2.4 Quadro de Dúvidas

Nas Reuniões Diárias, o Time de Desenvolvimento poderá citar algum problema de processo do *Big Data* que acontece na empresa, dúvidas que ainda não foram sanadas, e ainda comunicar o Scrum Master se o Sprint está atingindo o objetivo esperado.

As dúvidas existentes, seja do Sprint a ser tratado, do Product Backlog ou ainda de algum ponto específico do processo que ainda não foi citado, deverão estar descritas no Quadro de Dúvidas.

O Quadro de Dúvidas também é um novo artefato criado para atendimento da proposta. Seu objetivo é consolidar todas as dúvidas que irão surgindo, seja durante uma Sprint de aprendizagem, seja durante o desenvolvimento de um projeto.

O Quadro de Dúvidas será composto por 3 colunas: Nova, Resolvida, Pendente.

- Nova: nesta coluna constará todos os itens novos, não analisados pelo P.O. O analista do Time de Desenvolvimento escreverá a dúvida seguida de seu nome e a colocará no Quadro de Dúvidas, coluna “Nova”. O P.O. (Especialista) analisará a dúvida, se ele perceber que a dúvida é pertinente a etapa da Sprint, ele a responderá durante o desenvolvimento da Sprint.
- Resolvida: Dúvida já respondida em Sprint anterior. Se a dúvida já tiver sido sanada em outras Sprints, ela será movida para a coluna “Resolvida” e o P.O. indicará em qual item do Dicionário de Dados que ela se encontra, porém, ele pode auxiliar no entendimento da dúvida, mas, sem impactar o desenvolvimento da Sprint em questão.
- Pendente: Dúvida que ainda não foi respondida pelo P.O. Se a dúvida não tiver sido sanada, porém, referir-se a um item do Product Backlog, a dúvida constará na coluna “Pendente”. Senão, continuará na coluna “Nova”.

O Quadro de Dúvidas poderá ser tanto em *software* como físico, sendo ele, um quadro branco, uma apresentação projetada na parede, um *software* já existente que possa ser customizado, etc.

A seguir encontra-se o exemplo do Quadro de Dúvidas:

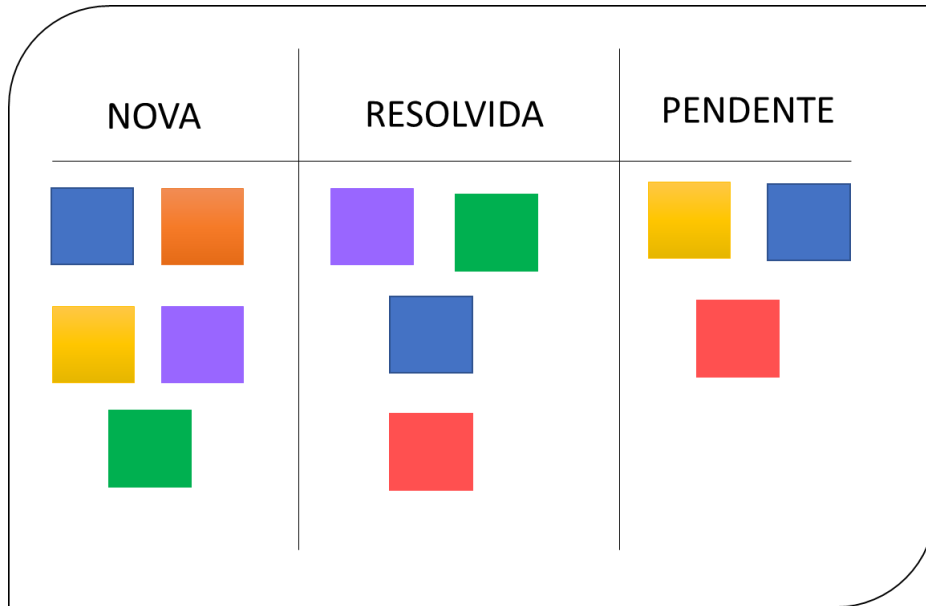


Figura 3 – Exemplo Quadro de Dúvidas

Fonte: Criado pelo autor

4.1.2.5 Desenvolvimento da Sprint de natureza não convencional

Durante o processo de desenvolvimento do projeto, isto é, o estudo das etapas do projeto *Big Data*, o P.O. poderá auxiliar o Time de Desenvolvimento, explicando etapas que foram customizadas para a empresa, concedendo dicas de realização de etapas de forma mais assertivas e rápidas e também lecionando tarefas mais complexas. Enquanto o Scrum Master garantirá que o *framework* Scrum seja seguido corretamente.

Após o decorrer das duas semanas, acontecerá a Revisão da Sprint, onde os integrantes do Time de Desenvolvimento juntamente com o Product Owner chegarão a um acordo sobre o aprendizado e analisarão se o tempo da Sprint foi suficiente.

A retrospectiva da Sprint também deverá ocorrer e o Time de Desenvolvimento inspecionará a si próprio e indicará um plano para melhorias a serem aplicadas na próxima Sprint.

4.1.2.6 Lista de Sugestões

Ainda durante a retrospectiva da Sprint, se o P.O. achar necessário, cada participante do Time de Desenvolvimento fará uma lista contendo sugestões de temas a serem aprendidos ou até mesmo dicas úteis para os outros membros da equipe.

Esta lista, não necessariamente, precisa conter o nome do membro da equipe, isto é, ela poderá ser anônima. E seu envio será feito via sistema, folha impressa, ou colocado em uma caixa. O importante é que os dados essenciais sejam preenchidos.

Todas as listas deverão ser enviadas para o P.O. para que ele selecione os 5 itens, que na sua visão são mais relevantes para o aprimoramento e análise da equipe.

A equipe realizará um Planning Poker para ordenar as sugestões/dicas concedidas e o P.O. as anotará para utilização no próximo Sprint. Caberá ao P.O. também repassar as sugestões/dicas recebidas ao Scrum Master para que ele possa garantir sua utilização.

As próximas Sprints seguirão o mesmo processo da primeira Sprint, a única diferença será o conteúdo dos seus Sprints Backlogs.

4.1.3 Comparativo gráfico

Com a utilização de Sprints de aprendizagem, pode-se aprimorar o *software* desenvolvido, pois os desenvolvedores (Time de Desenvolvimento) possuirão mais conhecimento técnico e poderão fazer melhores escolhas. Portanto, antes da execução dos Sprints, deve-se quantificar/qualificar alguns critérios nas entregas realizadas, para que se possa comparar com o próximo projeto, ou seja, pós Sprint de aprendizagem.

Os critérios essenciais para a comparação seriam:

- Percentual de aceitação do usuário;
- Utilização do desenvolvimento pelo usuário – este item nos certificará que o usuário está confiando mais nos dados entregues do que na sua intuição;

- Nível de capacitação da equipe;
- Custo – aquisições desnecessárias de tecnologia – redução de custo;
- Tempo de desenvolvimento – importante validar se houve diminuição nos retrabalhos da equipe;

O gráfico de comparação pode ser realizado via *softwares* de operações financeiras e contábilísticas (como Excel), *softwares* próprios para métodos ágeis, escritos em lousa ou outro material utilizado durante as Sprints de aprendizagem.

Exemplo:

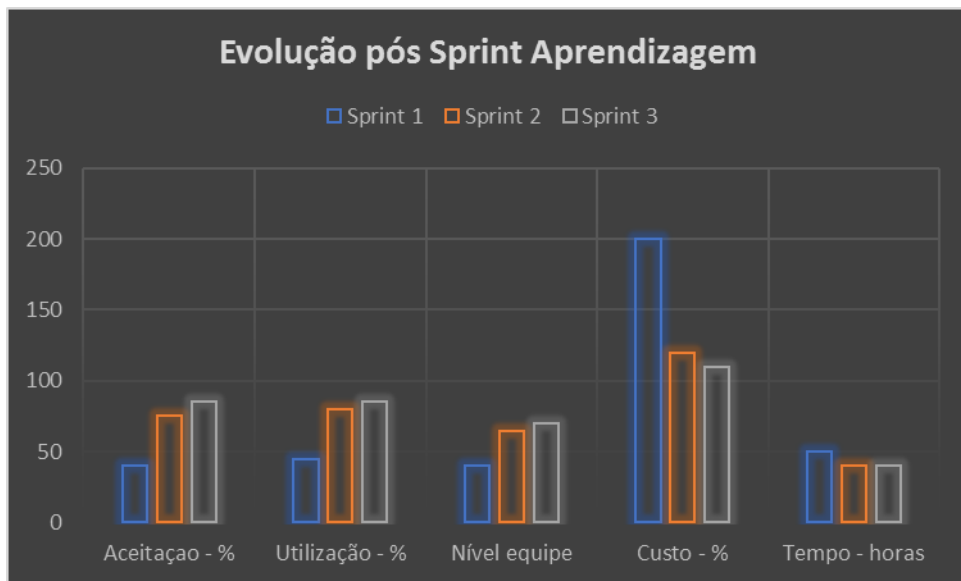


Figura 4 – Exemplo de comparativo gráfico das Sprints - antes, durante e depois da utilização de Sprints de aprendizagem

Fonte: Criado pelo autor

4.1.4 Resumo do processo final

Abaixo segue quadro resumo do processo:

Fase	Processos	Papéis
Inicial	1. Entender o problema	Product Owner
	2. Criar Visão do Projeto	Product Owner
	3. Criar Priorização do Product Backlog	Development Team, Product Owner
Planejamento e Estimativas	4. Criar User Stories	Product Owner
	5. Aprovar User Stories	Development Team, Product Owner
	6. Criar Tarefas	Development Team
	7. Estimar Tarefas	Development Team

	8. Criar Sprint Backlog	Development Team
Implementação	9. Preencher dicionário de conhecimento com as informações iniciais	Development Team
	10. Conduzir Standup Diário	Scrum Master
	11. Preenchimento do Quadro de Dúvidas	Development Team
	11. Manter Priorização do Product Backlog	Product Owner
Revisão e Retrospectiva	12. Complementar as informações aprendidas no dicionário de conhecimento	Development Team
	13. Validar o Sprint	Development Team, Product Owner
	14. Realizar Revisão do Sprint	Development Team, Product Owner e Scrum Master
	15. Realizar Retrospectiva do Sprint	Development Team, Product Owner e Scrum Master
	16. Criar lista de sugestões e definir as melhores	Development Team
Finalização	17. Anotar nas documentações da Sprint as melhores sugestões e garantir utilização na Sprint consecutiva	Product Owner, Scrum Master
	18. Retrospectiva do Projeto	Development Team, Product Owner e Scrum Master

Tabela 3 – Resumo Processo

Fonte: Criado pelo autor

5 CONSIDERAÇÕES FINAIS

Através dos estudos realizados nesta pesquisa foi possível concluir que uma das principais características das metodologias ágeis é que elas são adaptativas. Por este motivo, elas adaptam novas funcionalidades que podem surgir durante o desenvolvimento de um projeto, ao invés de serem preditivas e analisar, antecipadamente, tudo o que pode acontecer no decorrer do projeto. Portanto, para tratar sistemas que estão em constantes mudanças é recomendável que se utilize uma metodologia ágil.

Apesar do intenso crescimento do uso de metodologias ágeis nos dias atuais, ainda não é possível mensurar casos de sucesso em grandes e decisivos projetos. Quanto maior for a utilização das metodologias ágeis nas organizações, melhores serão os resultados de vantagens, desvantagens, riscos, procedimentos e conhecimento necessário da equipe para que ele seja adotado nas organizações. Logo, o maior desafio na utilização de metodologias ágeis é encontrar formas de acabar com seus pontos fracos.

O Scrum é um *framework* para organizar e gerenciar trabalhos complexos. Ele tem sido muito utilizado para gerenciar o desenvolvimento de sistemas por permitir imprevisibilidades dentro de um projeto. Por outro lado, o Scrum não possui um processo padronizado, ou seja, uma sequência de etapas que garantirá que o *software* será produzido dentro do prazo, do orçamento, com alta qualidade e de acordo com o que foi solicitado pelo usuário.

O *framework* Scrum permite que se adicionem vários processos e técnicas particulares de engenharia e gestão que sejam relevantes para a realidade de quem o utiliza. Com isso, se tem uma versão de Scrum exclusiva. Por isso, utilizar a metodologia ágil, Scrum, apenas aprimora a gestão do projeto, todavia os problemas de engenharia de *software* continuam.

Uma das áreas em que se encontram muitos problemas de engenharia de *software* é aquela nas quais são realizados projetos envolvendo o tratamento de *Big Data*. Atualmente, a competitividade entre as companhias de bens de consumo tem exigido um crescimento para que se entenda melhor o cliente. Por esta razão, o *Big Data* está muito em alta e as empresas que conseguirem aplicar corretamente este enorme mundo de dados que se apresenta, seguramente terão dados essenciais

para conseguir vantagens competitivas significativas. Porém, não há muitos profissionais qualificados para desenvolver este tipo de projeto e, também, a possibilidade de inclusão de profissionais com pouco ou nenhum conhecimento.

Por este motivo, este trabalho se propôs a criar uma proposta de utilização de Sprints de aprendizagem de projetos *Big Data* com apoio nas técnicas e ferramentas do Scrum. A proposta descrita é apenas um estudo conceitual que foi criado a partir de diversas fontes bibliográficas. Entretanto, não foi realizada a sua aplicação prática.

A proposta de aprendizagem de projetos *Big Data* com a utilização do *framework* Scrum foi realizada definindo os papéis, artefatos e eventos necessários a serem aplicados. O objetivo era manter as principais regras do Scrum, mas também adicionar as necessidades particulares de um projeto desta natureza.

Os objetivos do projeto foram atingidos. Foi explicado como a metodologia ágil Scrum será aplicada para aprendizado de projetos *Big Data* a fim de melhorar problemas de engenharia de *software*.

Mostrou-se não apenas a importância do *Big Data* nos dias de hoje, mas também, como é possível ter uma equipe heterogênea em relação a seus cargos e conhecimentos, utilizar uma metodologia ágil no desenvolvimento de projetos, aprender os artefatos e eventos desta metodologia durante Sprints educacionais e aprimorar o processo no decorrer dos projetos.

De acordo com o trabalho desenvolvido sobre Sprints de aprendizagem para desenvolvimento de projetos *Big Data*, embora o processo descrito seja apenas teórico e ainda não aplicado, é esperada que esta proposta seja utilizada, posteriormente, como base para uma solução prática em um cenário real de desenvolvimento de *software*, essencialmente em projetos de *Big Data*.

O ponto de partida para dar continuidade a esta pesquisa é aplicar em um ambiente real de desenvolvimento de *Big Data* o processo de Sprints educacionais desenvolvido neste trabalho. Isso permitirá não somente a utilização do processo, mas também uma possível adaptação no processo para que ele seja utilizado no dia-a-dia em empresas.

Apesar o processo ter sido desenvolvido inicialmente para *Big Data*, outros trabalhos futuros também podem adaptá-lo para outros tipos de *software*, realizando alterações que sejam pertinentes ao *software* a ser aprendido.

O estudo realizado conclui que os consumidores estão cada vez mais exigentes, o que aumenta a pressão entre as empresas e as torna cada vez mais competitivas e as obrigam a reagir de forma rápida, ou seja, esperam que as mudanças de *software*, criação de novas funcionalidades e inovações sejam realizadas cada vez em um menor período de tempo. Portanto, a utilização de metodologias ágeis nas empresas está começando a ser implementada, assim como o *Big Data*, e as empresas que conseguirem utilizar esse enorme universo de dados que é gerado diariamente em conjunto com o *framework* Scrum, certamente conseguirão ter melhores resultados.

REFERÊNCIAS

- ALVAREZ, Guilherme Martins. **Criação e utilização de uma base de dados orientada a grafos: Um estudo de caso sobre rede social**. 2013. Disponível em: <https://riuni.unisul.br/bitstream/handle/12345/664/109625_Guilherme.pdf?sequence=1&isAllowed=y>. Acessado em: 05 nov. 2017.
- BECK, et al. **Manifesto para Desenvolvimento Ágil de Software**. Disponível em: <<http://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 23 jan. 2017.
- DISNER, Daniel da Silva. **Mineração de dados para obtenção de conhecimento em Big Data**. 2014. Disponível em: <<http://aberto.univem.edu.br/handle/11077/996>>. Acessado em: 12 ago. 2017.
- EREVELLES, S. et al. **Big Data consumer analytics and the transformation of marketing**. 2015. Disponível em: <https://www.researchgate.net/publication/279910868_Big_Data_Consumer_Analytics_and_the_Transformation_of_Marketing>. Acesso em: 15 jul. 2017
- GIACALON, M. e SCIPPACERCOL, S. **BIG DATA: ISSUES AND AN OVERVIEW IN SOME STRATEGIC SECTORS**. 2016. Disponível em: <https://www.jaqm.ro/issues/volume-11,issue-3/pdfs/0_MA_SE_.pdf>. Acesso em 11 de set. 2017.
- NESELLO, Priscila; FACHINELLI, Ana Cristina. **BIG DATA: O NOVO DESAFIO PARA GESTÃO**. 2014. Disponível em: <<http://www.inteligenciacompetitivarev.com.br/ojs/index.php/rev/article/view/76>>. Acesso em 09 de ago. 2017.
- PEREIRA, Vanessa Alves da Silva. **Big Data: um estudo em gestão empresarial**. 2015. Disponível em: <<http://pantheon.ufrj.br/bitstream/11422/169/1/Big%20data%20-%20um%20estudo%20em%20gest%C3%A3o%20empresarial.pdf>>. Acesso em 15 ago. 2017.
- SABBAGH, Rafael. **Scrum: Gestão ágil para projetos de sucesso**. São Paulo: Casa do Código, 2013.
- SANTOS, Rafael. **Conceitos de Mineração de Dados na Web**. 2009. Disponível em: <<http://www.lac.inpe.br/~rafael.santos/Docs/WebMedia/2009/webmedia2009.pdf>>. Acesso em 20 set. 2017.

SCHWABER, Ken; SUTHERLAND, Jeff. **Guia do ScrumMR: Um guia definitivo para o Scrum: As regras do jogo**. 2016. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acesso em 23 jan. 2017.

SOARES, Michel dos Santos. **Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software**. 2004. Disponível em: <<http://www.periodicosibepes.org.br/index.php/reinfo/article/view/146/38>>. Acesso em 30 jan. 2017