



DIFICULDADES DE IMPLANTAÇÃO DA METODOLOGIA ÁGIL SCRUM

IMPLEMENTATION DIFFICULTIES OF SCRUM AGILE METHODOLOGY

Gabriel Inácio dos Santos - gabrazil03@hotmail.com

Daniela Gibertoni - daniela.gibertoni@fatectq.edu.br

Faculdade de Tecnologia de Taquaritinga (FATEC) – São Paulo – Brasil

RESUMO

Conforme as empresas necessitam de diferentes métodos para a criação de novos produtos, novas abordagens são requeridas para que as mesmas possam atender suas necessidades. Atualmente inúmeras metodologias estão sendo utilizadas para o desenvolvimento de software com o intuito de satisfazer o cliente da melhor maneira possível. A metodologia ágil Scrum é bastante utilizada para o desenvolvimento de software por ser flexível em um ambiente de constantes mudanças. O Scrum consiste em criar uma equipe composta por três papéis fundamentais e dividindo entre eles as tarefas, a fim de obter rapidez na entrega de um produto. Apesar de parecer um método simples de se utilizar este trabalho tem como objetivo identificar as dificuldades que a implementação da metodologia ágil Scrum traz quando empregada, do mesmo modo apresentando um estudo de caso, relatando como foi o planejamento e desenvolvimento de uma aplicação mobile e quais foram as dificuldades encontradas com a utilização da metodologia ágil Scrum durante o desenvolvimento do projeto.

Palavra-chave: Metodologia ágil, Metodologia Scrum, Desenvolvimento de software.

ABSTRACT

As companies need different methods for creating new products, new approaches are needed so that companies can meet their needs. Currently many methodologies are being used for the development of software in order to satisfy the customer in the best possible way. The agile Scrum methodology is widely used for software development because it is flexible in an environment of constant change. Scrum consists of creating a team composed of three fundamental roles and dividing tasks among them in order to obtain speed in the delivery of a product. Although it seems a simple method to use this work, aims to identify the difficulties that the implementation the agile Scrum methodology brings when implemented, likewise presenting a case study, reporting how was the planning and development of a mobile application and which were the difficulties encountered with the use of agile Scrum methodology during the development of the project.

Keywords: Agile Methodology, Scrum Methodology, Software Development, Difficulties encountered.



1. INTRODUÇÃO

As metodologias ágeis, atualmente, estão sendo cada vez mais utilizadas para o desenvolvimento de software, elas são uma alternativa para as metodologias tradicionais, conhecidas também como metodologias pesadas. A maioria das metodologias ágeis não possuem nada de novo, sua diferença entre as metodologias tradicionais são o enfoque e os valores, que são voltados para pessoas e não para processos, visando gastar menos tempo com documentações e mais com a implementação.

Uma característica das metodologias ágeis é que elas são adaptativas ao invés de serem preditivas. Com isso, elas se adaptam a novos fatores decorrentes do desenvolvimento do projeto, ao invés de procurar analisar previamente tudo o que pode acontecer no decorrer do desenvolvimento (SOARES, 2004).

O Scrum é uma metodologia extremamente ágil e flexível, que tem como objetivo definir um processo iterativo e incremental onde pode ser aplicada a qualquer atividade. A metodologia Scrum pode ser aplicada em projetos grandes ou pequenos, onde seu objetivo principal é conseguir uma avaliação do ambiente em desenvolvimento, adaptando-se a novas necessidades. No ambiente de desenvolvimento de software, onde requisitos estão em constante mudança, o Scrum é uma grande solução (LOPES, 2014).

Bissi (2007) descreve “A Metodologia Scrum apenas estabelece conjuntos de regras e práticas de gestão que devem ser adotadas para garantir o sucesso de um projeto”.

O Scrum apresenta uma abordagem empírica que aplica algumas ideias da teoria de controle de processos industriais para o desenvolvimento de softwares, reintroduzindo as ideias de flexibilidade, adaptabilidade e produtividade (SOARES, 2004). Mas será que a implantação do Scrum é tão fácil assim?

Este trabalho tem como objetivo identificar as dificuldades que a implementação da metodologia ágil Scrum traz quando efetuada, com base em um estudo de caso.

2. METODOLOGIA ÁGIL SCRUM

Scrum é um método ágil que controla a evolução de um produto de maneira eficaz. Ele pode ser definido como um conjunto de papéis, eventos e regras associados a uma equipe, otimizando seu trabalho. A metodologia *Scrum* busca trabalhar todos os elementos de uma



equipe afim de produzir *softwares* de uma forma flexível em um ambiente em constante mudança com foco na entrega do maior valor do negócio, em menor tempo possível (LOPES, 2014).

A utilização do Scrum é indicada quando possíveis mudanças nos requisitos possam acontecer durante seu processo de desenvolvimento. Reúne atividades de monitoramento e *feedback*, em geral, reuniões rápidas e diárias com toda a equipe, visando à identificação e correção de quaisquer deficiências e/ou impedimentos no processo de desenvolvimento (SCHWABER; BEEDLE, 2002).

O nome *Scrum* foi dado devido ao esporte chamado *Rugby* onde o trabalho de equipe é fundamental. Nesse esporte um time pontua quando a bola cruza a linha do gol e toca o chão, para isso é necessária que o time realize uma intensa troca de passes, para evitar serem derrubados por outros jogadores e atingirem seu objetivo, caso o jogador seja derrubado ele deve soltar a bola e a jogada se reinicia. Cada jogada se inicia quando um *scrum* é realizado, que é uma parede de força entre jogadores. A ideia é que todos se unirem e unindo suas habilidades e força para tralharem em conjunto e em harmonia para que consigam recuperar a bola. O time todo deve trabalhar junto para que a equipe possa pontuar.

2.1 Papéis no gerenciamento de projeto com *scrum*

O time *scrum* é composto por três papéis fundamentais, o *Product Owner*, *Scrum Master* e o Time de Desenvolvimento, esses papéis são necessários para a entrega de *softwares* com alto valor agregado.

2.1.1 *Product Owner*

O *Product Owner* é “dono do projeto”. É o responsável por maximizar o valor do produto e do trabalho da equipe de desenvolvimento. Ele deve possuir uma visão do produto em vários níveis (LOPES, 2014).

O *Product Owner* é o único que tem autorização e pode tomar decisões em relação ao produto e a equipe de desenvolvimento.

Sendo o responsável por decidir quais recursos e funcionalidades serão construídos e qual a ordem que eles devem ser feitos, o *product owner* tem a responsabilidade de manter e



comunicar a todos os outros participantes uma visão clara do que a equipe *scrum* está buscando alcançar no projeto (BISSI, 2008).

2.1.2 *Scrum Master*

O *Scrum Master* é a pessoa responsável por garantir que o *Scrum* seja entendido e aplicado, sendo assim o mesmo deve ter conhecimento de todos os processos do *Scrum*. O *Scrum Master* faz isso para garantir que o Time *Scrum* adere à teoria, práticas e regras da metodologia *Scrum*. O *Scrum Master* é um servo-líder para o Time *Scrum*. (SCHWABER; SUTHERLAND, 2014).

Responsável por ajudar a todos os envolvidos a entenderem os valores, princípios e práticas da metodologia, o *Scrum Master* deve conhecer muito bem o *scrum*, executando a liderança do processo e ajudando a equipe a desenvolver sua própria abordagem do mesmo.

O *Scrum Master* interage tanto com a Equipe *Scrum*, como com os Clientes e o Gerente durante o projeto (CRISTIANE; SILVEIRA, 2010).

2.1.3 Equipe de desenvolvimento

É responsável pelo desenvolvimento do produto e é formada por desenvolvedores. Consiste em profissionais que realizam o trabalho de entregar uma versão usável que potencialmente incrementa o produto “Pronto” ao fim de cada *sprint* (LOPES, 2014).

No *scrum* todos os membros da equipe têm o mesmo cargo de desenvolvedores, todos devem ter a capacidade para entender o trabalho efetuado em qualquer parte da implementação, é perfeitamente normal que alguns elementos tenham um domínio maior e que estejam mais acostumados a fazerem determinadas tarefas.

A Equipe de Desenvolvedores é composta de fato pelas pessoas que constroem o projeto, e quem decide como as coisas serão feitas é a equipe e não o gerente ou qualquer outra pessoa. A ideia principal é que o time se auto organize para determinar a melhor maneira de realizar o trabalho para atingir a meta estabelecida pelo *product owner*. A equipe deve conter entre três a dez integrantes, para que não tenha limitações de habilidades por ser pequena de mais, e não devem ser tão grandes dificultando serem gerenciadas (CYSNEIROS, 2007).



2.2 Termos técnicos

2.2.1 *Product backlog*

Product backlog é a lista com todas as tarefas necessárias para o desenvolvimento do projeto que avança de acordo com o desenvolvimento do mesmo. O *product owner* com a ajuda do *scrum master* deve priorizar os itens da lista de tarefas para garantir a produtividade.

Costa (2013) descreve “É o local onde estão ordenados e documentados todos os requisitos funcionais do produto, tanto os implementados como os por implementar. Artefato de responsabilidade do *Product Owner* e gerido com a ajuda do *Scrum Master*, deve estar sempre visível e disponível para consulta da equipe ou de outros *stakeholders* interessados.”

2.2.2 *Sprint*

Lopes, 2014 diz que “O coração do *Scrum* é a *sprint*. Os projetos que utilizam o *Scrum* são divididos em ciclos chamados de *sprint*, os quais correspondem a iterações.”

Sprints são reuniões onde alguns itens selecionados do *product backlog* são construídos e entregues para prosseguir com o projeto. Para planejar as *sprints* o ideal é que as mesmas tenham um limite de duração fixo. Quando o horizonte da *Sprint*, é muito longo, a definição do que será construído pode mudar, a complexidade pode aumentar e o risco pode crescer. Normalmente é feita a cada duas ou quatro semanas, uma nova *sprint* só é iniciada quando a anterior for terminada (SCHWABER., SUTHERLAND 2013).

2.2.3 *Sprint planning meeting*

O planejamento da *sprint* é feito com base no tamanho da equipe para ser definido um número de funcionalidades que possam ser construídas durante o tempo da mesma, onde os próximos itens que serão desenvolvidos seguem a ordem e a importância definida pelo *product owner*. O *product owner* pode verificar se há necessidades de mudanças e se as mesmas, são inseridas novamente no *backlog* conforme sua prioridade. O processo se repete até que o produto final estiver pronto e com todas as mudanças adicionadas.



A reunião de planejamento da *Sprint* (*Sprint planning meeting*) possui um *time-box* com no máximo oito horas para uma *Sprint* de um mês de duração. Para *Sprints* menores, este evento é usualmente menor (SCHWABER; SUTHERLAND, 2013).

2.2.4 *Daily scrum*

Daily Scrum são reuniões diárias de curto tempo de duração para que os membros possam harmonizar o trabalho. Durante a reunião a três questões chaves para serem respondidas:

- O que foi feito desde a última *Daily Scrum*?
- O que será feito para a próxima *Daily Scrum*?
- Existe algum obstáculo que bloqueia a realização da tarefa?

Esta reunião é utilizada pelo *Scrum Master* para perceber a direção e a velocidade da equipe na *Sprint* que está a decorrer, bem como para os elementos da equipe notarem o estado das tarefas e as dificuldades encontradas para resolver (COSTA, 2013).

2.2.5 *Sprint review meeting*

O objetivo da *sprint review* é validar e ajustar o produto que está sendo construído, verificando se o que foi feito está conforme o esperado, ou seja, apresentar tudo o que foi feito durante a *sprint*. É nesta fase que podem surgir necessidades de mudanças e o momento em que o *product backlog* é atualizado.

No último dia do *sprint*, é incrementado ao *software* todo o trabalho efetuado no mesmo e é apresentado às partes interessadas para avaliação (FERRAZ, 2016).

3. ESTUDO DE CASO

Nesta sessão é apresentado um estudo de caso, desenvolvido na Faculdade de Tecnologia de Taquaritinga, na matéria Laboratório de Engenharia de *Software*, onde foi proposto em grupos de quatro integrantes desenvolver um *software*, *site* ou uma aplicação *mobile* onde a metodologia utilizada para o desenvolvimento foi a *Scrum*.



No desenvolvimento do *software*, os integrantes do grupo apresentaram suas ideias, e uma foi escolhida. O grupo optou por desenvolver uma aplicação *mobile* na área da saúde, denominado “*Blood is life*”, um aplicativo que cadastra doadores de sangue, bem como os hospitais que recebem doações, e a medida em que um determinado tipo sanguíneo é solicitado, o doador correspondente ao tipo sanguíneo é notificado e informado aonde e para que será utilizado o sangue que está sendo doado.

3.1 Papéis

O integrante no qual a ideia do aplicativo foi escolhida, além de desenvolvedor também foi o *product owner*, sendo ele quem informou as funcionalidades que o aplicativo deveria conter, aceitando sugestões dos outros integrantes da equipe e qual seria a ordem de implementação dessas funcionalidades, priorizando os itens do *product backlog*.

O professor, por ter um grande conhecimento sobre *scrum*, e como funciona essa metodologia, atuou para todos os grupos como *scrum master*, onde o *product owner* mostrava a ideia e informava o que seria feito durante a *sprint*. O *scrum master* tirava dúvidas da equipe de desenvolvimento, caso algo não tivesse sido entendido por algum dos integrantes.

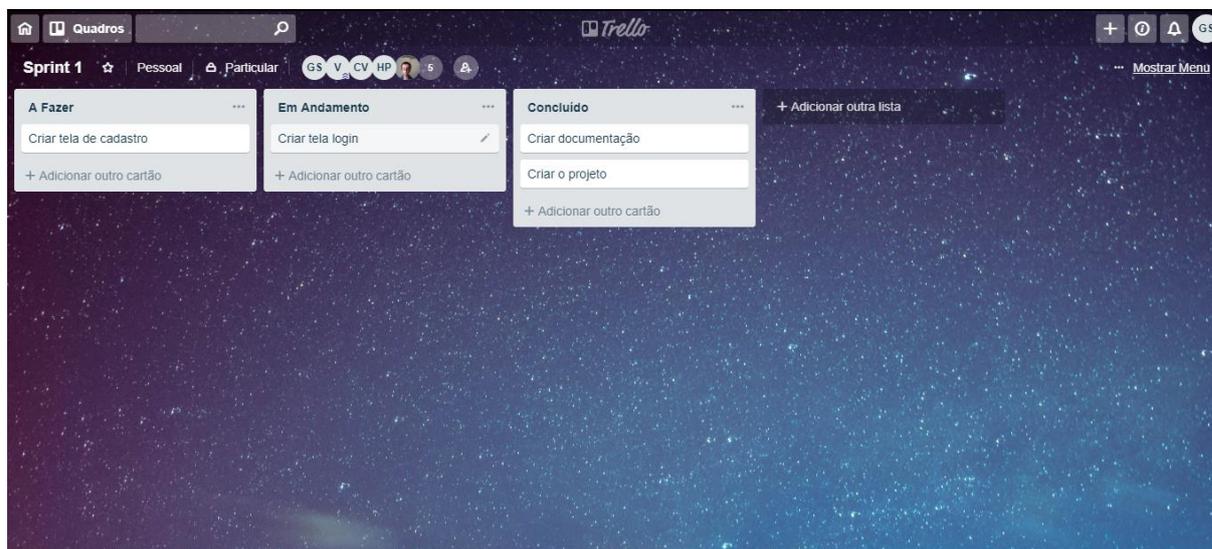
O time de desenvolvimento foi composto pelos quatro integrantes do grupo entre eles o *product owner* que também foi um dos desenvolvedores, pois no time *scrum* todos tem o mesmo cargo todos são desenvolvedores.

As *sprints* foram marcadas a cada duas semanas, onde os itens a serem feitos no *product backlog* eram entregues para o professor avaliar se realmente havia sido feito o que foi proposto.

3.2 Relato do caso

Após a definição do projeto que o time de desenvolvimento iria desenvolver, foram criadas as funcionalidades que deveriam conter no aplicativo. Assim o *product owner* organizou a *product backlog*, priorizando as funcionalidades de acordo com o que era necessário para prosseguir com o desenvolvimento do aplicativo. De acordo com a imagem 1, e cada membro ficou responsável por uma parte.

Imagem 1 – Primeira sprint formada das funcionalidades priorizadas



Fonte: Trello.com

A *daily scrum* foi combinada entre os membros do time a ser realizada diariamente, as vinte e duas horas, onde cada um dos membros informa o que estava sendo feito e se havia algum bloqueio que estava o impedindo de prosseguir, assim o time de desenvolvimento sabia como estava o andamento do projeto.

Na semana antes da *sprint*, a *daily sprint* era realizada com o *scrum master* para saber como estava o projeto até o momento, onde eram feitas três perguntas, onde cada membro mostrava o que estava sendo feito e o que estava bloqueando para prosseguir, o professor auxiliava com possíveis soluções do que poderia ser feito para quebrar o bloqueio e dar continuidade ao desenvolvimento do projeto.

No dia da *sprint*, reunia-se o time de desenvolvimento e o *scrum master*, onde era mostrado o que havia sido combinado para entregar e o que havia sido entregue. Após a revisão do que havia sido prometido e o que foi feito, o time informava para o *scrum master* o que seria entregue na próxima *sprint*.

O projeto teve um tempo definido para ser entregue, pois seria apresentado em sala para os demais grupos e para que o professor pudesse avaliar. Cada grupo relatou a experiência adquirida com a utilização da metodologia *scrum*. E tendo visto na prática como essa metodologia funciona, no final, o professor pode avaliar o desenvolvimento das equipes, dando críticas construtivas de como melhorar ainda mais o projeto caso algum grupo estivesse com intenção de disponibilizar para outras pessoas o projeto final.



3.2.1 Ferramentas utilizadas

Foi utilizada a ferramenta Trello para a organização do *product backlog* e da criação das *sprints* do projeto. O Trello é uma ferramenta que serve para organizar projetos, através de quadros compartilhados com membros de um grupo, de início ela informa o que está sendo trabalhado, quem está trabalhando em determinada parte, e o que está sendo processado.

No Trello existem três tabelas, a tabela a fazer, a em andamento e a concluído.

Na tabela a fazer é listado tudo o que dever ser feito durante um determinado tempo, os itens da tabela a fazer podem ser arrastados para a tabela em andamento, facilitando para os membros da equipe saberem o que já está sendo feito. Após o termino do item que estava em andamento ele passa para a tabela concluída.

Para a realização da *daily scrum* foi utilizado a ferramenta Skype, para que o grupo se reunisse e conversasse sobre o andamento do projeto.

O Skype serve para realizar tarefas em grupo independente da distância. A ferramenta possui diversos recursos como, SMS, chamadas de voz e vídeo, permitindo assim compartilhar experiências de uma maneira prática com outras pessoas em qualquer lugar.

3.2.2 Dificuldades encontradas

Uma das maiores dificuldades encontradas foi realizar a *daily scrum* todos os dias, devido a problemas externos de cada um da equipe, e assim algumas reuniões foram realizadas por dois ou três integrantes, fazendo com que a velocidade de desenvolvimento do aplicativo fosse reduzida.

A IDE utilizada para o desenvolvimento do aplicativo, possuía requisitos altos para seu funcionamento, o que dificultou as divisões das tarefas, pois apenas uma das máquinas atendia os requisitos mínimos para utilização da mesma. Foi adquirida uma nova máquina que também atendia os requisitos mínimos da IDE, assim as divisões das tarefas puderam ser realizadas de maneira que não sobrecarregasse a equipe.

A pouca familiaridade da equipe com desenvolvimento de aplicações *mobile* atrasou o início do projeto, pois foi necessário aprender primeiro como utilizar a IDE, para que se iniciasse o mesmo, onde a primeira tarefa era desenvolver todas as telas feitas na prototipagem do aplicativo e suas funcionalidades.



O *back-end* do aplicativo foi desenvolvido na linguagem de programação Java com o *framework Spring Boot* que foi recomendado pelo *Scrum Master* para facilitar o desenvolvimento do aplicativo. O desenvolvimento do projeto parou pela falta de conhecimento do *framework* utilizado, onde novamente foi necessário pesquisar e entender como utilizar o mesmo, com pesquisas e ajudas de pessoas com conhecimento sobre ele, foi possível utilizar o *Spring Boot* e dar continuidade ao projeto.

As máquinas utilizadas para o desenvolvimento do aplicativo com a utilização da IDE e do *Spring Boot* ao mesmo tempo, ficaram lentas e muitas vezes travavam na hora de rodar o aplicativo para realizar testes, atrasando o andamento do projeto.

3.3 Resultados

Esta sessão teve como objetivo relatar as características da metodologia *Scrum* e os problemas encontrados com sua utilização.

Apesar das dificuldades encontradas durante a implementação do *Scrum* pode-se notar a eficiência e a flexibilidade do método, pois ao longo do projeto ajustes eram feitos e com isso acarretava pequenas mudanças. O projeto teve seu término antes do prazo combinado para ser entregue, pois devido ao empenho e comprometimento da equipe em concluir todas as tarefas divididas entre si e a ajuda do *Scrum Master* com possíveis soluções para a conclusão dessas tarefas, o time *scrum* atingiu o objetivo e o projeto foi entregue com sucesso no dia previsto.

4. CONCLUSÃO

Grande parte das metodologias ágeis surgiram da necessidade de novas formas para que diferentes resultados fossem obtidos.

O aumento das disputas entre empresas de desenvolvimento de *software* faz com que estas sempre busquem novas formas para aumentar a produtividade, a redução no tempo de entrega do produto, a melhora na qualidade e a redução de custos. O objetivo principal dessas mudanças é manter a competitividade e a satisfação do cliente.

Assim, empresas começaram a utilizar novas formas de desenvolvimento de *softwares*, sendo elas as metodologias ágeis. O *Scrum* é uma dessas formas, ele indica eventos, papéis e



artefatos que, se utilizado de forma correta, é possível gerenciar uma equipe que possua um alto nível de comprometimento em relação ao trabalho a ser executado.

As vantagens da utilização do *Scrum* é a disponibilidade para fazer mudanças o tornando flexível, a rapidez e qualidade na entrega do produto final.

A conclusão final que se pode chegar com a utilização da metodologia ágil *Scrum* no desenvolvimento de *software*, é que apesar de ser uma metodologia simples de ser utilizada, para ser eficaz é necessário que a equipe *scrum* seja comprometida para a realização de um determinado trabalho, para que o objetivo em questão possa ser alcançado com sucesso. Atuando como desenvolvedor isto pôde ser comprovado por meio do estudo de caso apresentado.

REFERENCIAS

BASSI, D. L. F. Experiencias com desenvolvimento ágil. Tese de Mestrado, Instituto de Matemática e Estatística da Universidade de São Paulo. p. 38-41, mar. 2008.

BISSI, W. Scrum - Metodologia de desenvolvimento ágil. Revisão Bibliográfica, **Campo Dig.**, **Campo Mourão**, v.2, n.1, p.3-6, jan/jun. 2007.

COSTA, A. A., Planejamento e Gestão Concorrente das Equipas Scrum: Das Arquiteturas Lógicas. Tese mestrado, **UMinho**. p.17 – 21, out. 2013.

CRISTIANE, A. F.; SILVEIRA, H. M. Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum, Lean. **UNICAMP – Universidade Estadual de Campinas, FT – Faculdade de Tecnologia**, p.12-16, 2010.

FERRAZ, S. R. C., Recomendações para a adoção de praticas ágeis no desenvolvimento de software: estudo de casos. Tese de Mestrado, **UMinho**.

LOPES, C. F. S. Scrum para Ambientes de Software Distribuído: Análise Crítica e Estudo de Casos. p. 33-49, out. 2014.

SCHWABER, K; BEEDLE, M., Agile Software Development With Scrum. NJ: **Prentence Hall**, 2002.

SCHWABER, K.; SUTHERLAND, J. (2011). Guia do Scrum. Retrieved January 02, 2013. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 27 ago. 2018.

SOARES, M. S. Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. **INFOCOMP**, [S.l.], v. 3, n. 2, p. 8-13, nov. 2004.



SOARES, M. S. Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. **Revista Eletrônica de Sistemas de Informação**, v. 3, n. 1, p. 1-8, 2014.

CYSNEIROS, A. S. M. Estendendo o Scrum segundo as Áreas de Processo de Gerenciamento de Projetos do CMMI. v.2, p. 3-6, 2007.