Título Estudo de caso: aplicação do método *Scrum* a desenvolvimentos extremamente ágeis

Resumo

Durante o desenvolvimento de um software, com o emprego da metodologia ágil *Scrum*, mas com um fator de complexidade adiconal, um prazo extramamente curto para sua entrega, a metodologia que muito bem serviu a fábrica de software em outros projetos, mostrou-se deficiente em alguns aspectos. Para superar tais obstáculos buscaram-se adaptações ao método, estas adaptações serão objetos de um estudo de caso, onde se pretende analisar a eficiência da metodologia ágil *Scrum* para demandas que tenham prazos muito curtos. Buscar-se-á na bibliografia hoje existente se tais adaptações são saudáveis à metodologia *Scrum* e se ele poderia sofrer adaptações para casos da mesma natureza, podendo apresentar entre os resultados deste trabalho, propostas para melhoria da metodologia.

Palavras-chaves

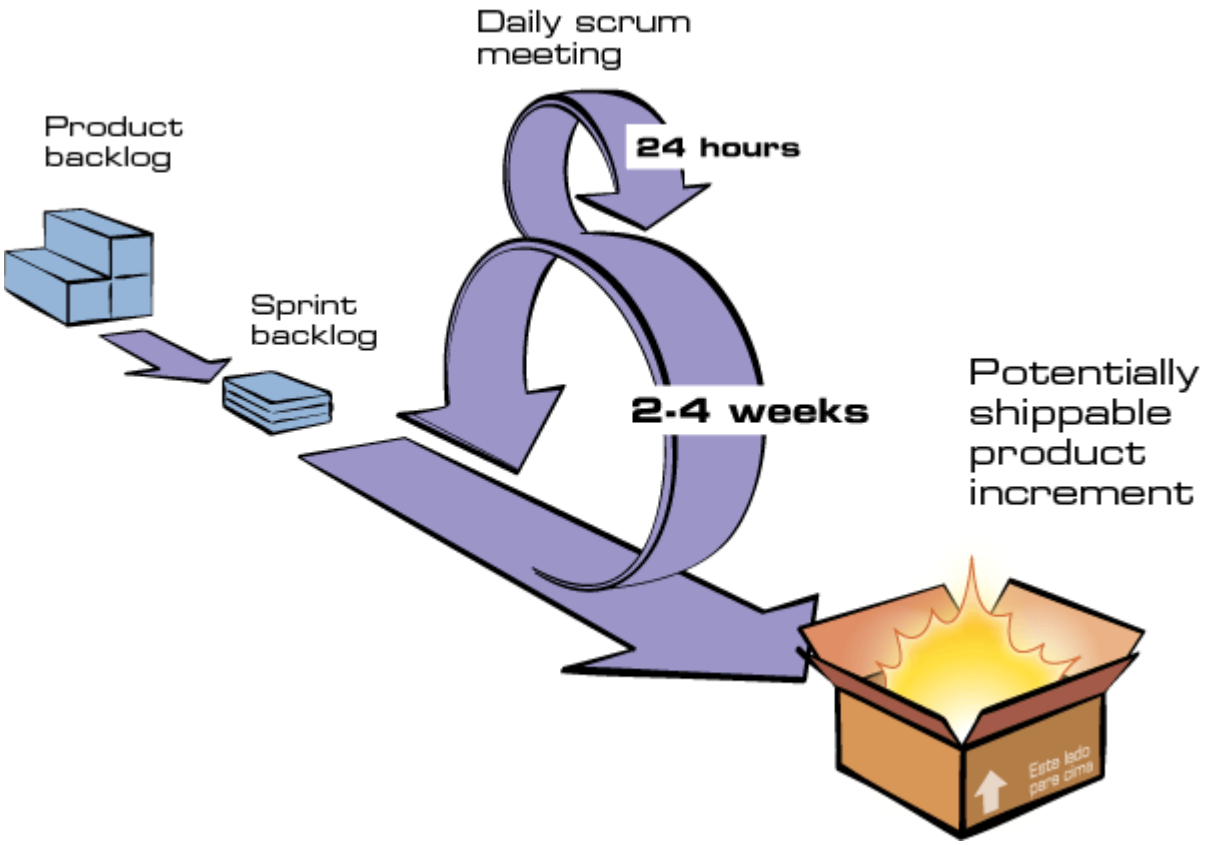
Método ágil; *Scrum*; Desenvolvimento; Software

Introdução e justificativa, com síntese da bibliografia fundamental

O *Scrum* é uma metodologia ágil que vem buscar uma solução para os problemas encontrados pelas empresas desenvolvedoras de software frente aos modelos tradicionais que tem como seu maior expoente a metodologia em Cascata (ou *Waterfall* em inglês), que faz todo o desenvolvimento do projeto completando todos os passos anteriores antes de seguir para a próxima etapa, o maior problema encontrado com os métodos tradicionais é o fato de que mudanças são necessárias e quanto mais tempo se demora em fazê-las, mais caro e demorado fica o projeto. Assim as metodologias ágeis, como o *Scrum*, buscam fazer entregas parciais do projeto, permitindo uma maior interação do cliente e, consequentemente, permitindo que alterações sejam detectadas antes das etapas finais e incorporadas mais rapidamente e com menor custo.

A metodologia *Scrum* tem seu nome inspirado numa jogada que acontece no esporte de nome *Rugby*, sendo uma forma de reinício do jogo e onde ambos os times se juntam e se empurram com o objetivo de ganhar a posse da bola, neste momento o time precisa saber trabalhar junto para alcançarem o objetivo proposto.

Figura 1: Ciclo *Scrum*



Conforme se observa da figura acima, cada ciclo de desenvolvimento da metodologia *Scrum* deve ser de 2 a 4 semanas, após o que deve haver algo entregável para que o cliente possa validar e ter a oportunidade de pedir melhorias ou mudanças no sistema, que está sendo desenvolvido, o mais cedo possível, diminuindo custos e tempo.

O método ainda prevê reuniões diárias (*Daily Scrum meetings*) logo no início do dia e que devem ser feitas com o time em pé e não podem passar de 15 minutos, nela cada um da equipe deve dizer o que fez no dia anterior, o que vai fazer no dia corrente e se está encontrando algum problema que precisa de ajuda.

Ainda é possível observar que a entrega ao final de um ciclo, também chamado de *Sprint*, é de uma parte do produto e não do produto todo, o que o diferencia dos métodos tradicionais, em especial o método *Waterfall*.

Mas, talvez o principal ponto de atenção da metodologia *Scrum*, o PO apenas participará do planejamento da *Sprint* e depois, ao findar o ciclo, da *Review* da *Sprint*, durante os demais dias ele deve estar junto com o Analista de Requisitos para preparar quais são os produtos que poderão entrar na próxima *Sprint*.

Assim deve ser a metodologia de desenvolvimento *Scrum*, toda a equipe trabalhando em conjunto para entregar algo de valor, no início esta metodologia era aplicada basicamente para o desenvolvimento de software, contudo hoje ela migrou para uso nos mais diversos projetos onde é necessária uma equipe unida em torno de um objetivo em comum.

Uma empresa de desenvolvimento de software, que comumente emprega a metodologia *Scrum* em seus projetos, e que vinha servindo perfeitamente às suas necessidades, se viu com dificuldades quando aportou uma demanda que tinha como premissa a entrega do produto em um prazo muito curto.

Foram necessárias adaptações na metodologia *Scrum* para que o projeto pudesse ser entregue dentro do prazo estabelecido. Seriam viáveis tais adaptações na metodologia ou estaremos observando um novo método para o atendimento de demandas com premissas semelhantes a enfrentada por aquela fábrica de software?

Jeff Stherland, juntamente com Ken Schwaber (2020), ambos co-desenvolvedores do framework *Scrum*, nas notas finais em "O Guia do *Scrum*" apontam que a metodologia, conforme apresentado no guia, é imutável.

Moraes, Borges e Okuyama (2013) também sustentam que a metodologia *Scrum* é capaz de agregar toda a equipe, ao mesmo tempo em que incorpora responsabilidade individual e compartilhamento de conhecimentos, mesmo não tendo verificado a necessidade de adaptações.

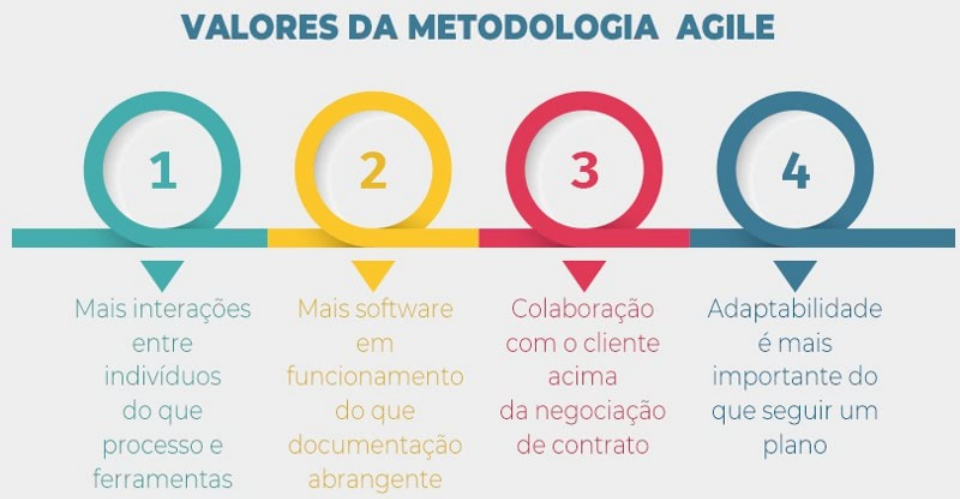
Contudo, algumas vezes precisamos parar de fazer o que estamos fazendo e rever a nossa metodologia nos indagando se seria possível fazer o que fazemos de uma forma melhor, o próprio autor Jeff Sutherland (2014) chama isso de "Inspeção e Adaptação".

Outros autores pesquisados como Ayed et. al (2014), que também estudaram a implantação da metodologia *Scrum* em projetos de desenvolvimento de softwares, destacaram as utilidades do *Scrum*, lembrando, contudo que adaptações foram necessárias para cada projeto alvo estudado.

Ao nos valermos dos valores e princípios contidos no manifesto ágil, e que são a base para o desenvolvimento de metodologias ágeis como o *Scrum*, sendo que seus fundadores (do *Scrum*) também são signatários do manifesto ágil, podemos destacar os seguintes pontos:

O manifesto ágil possui 4 valores, como podem ser observados na figura 2.

Figura 2: Valores do manifesto ágil



Fonte: SCHWABER, K. e SUTHERLAND, J. (2020).

Valendo destacar o valor número 4 que trata da adaptabilidade, apontando que ela é mais importante que seguir um plano, destacando a vantagem da adaptabilidade, caso o método não esteja sendo o mais eficaz e uma adaptação possa ser feita, ela será bem-vinda.

Quanto aos princípios que compõe o manifesto ágil, ele possui 12 princípios que podemos ver na figura 3.

Figura 3: Princípios do manifesto ágil

Fonte: SCHWABER, K. e SUTHERLAND, J. (2020).

Aqui se destacam o princípio de número 6, que aponta que o método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de uma conversa face a face, este princípio aponta para a ideia de conseguir transmitir as necessidades do cliente diretamente para os desenvolvedores, sem a necessidade de passarmos por um intermediário, o analista de requisitos, o que no caso em estudo foi muito mais eficiente.

E o princípio número 11 que indica que as melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis. Aqui temos um conflito com o framework *Scrum* que apresenta uma organização estanque, esta imutabilidade não se mostra aderente a este princípio do manifesto ágil, verificamos, no caso em estudo, que ao substituir a *Daily Scrum* por interações contantes durante todo o dia temos um exemplo de auto-organização.

O estudo de caso objeto deste projeto recai sobre a demanda de um desenvolvimento de software que aportou em uma fábrica de software que normalmente utiliza da metodologia ágil *Scrum* para o atendimento de tais demandas, contudo havia um fator complicador, o prazo para entrega era extremamente enxuto, e o método que sempre funcionou estava se mostrando o principal entrave no cumprimento do prazo. Toda a equipe envolvida, desde os desenvolvedores até os representantes da área de negócio se mostraram muito apreensivos e

iniciou-se uma busca por uma solução que pudesse contornar o problema, à época um colaborador havia assistido ao filme “Apollo 13: do desastre ao triunfo” (1995) e inspirado nas dificuldades interpretadas, incentivou aos demais envolvidos no projeto a, além de assistirem aquela produção artística, apresentarem ideias, sendo algumas delas eleitas e testadas naquele projeto em específico e que permitiu, ao final, o desenvolvimento e a entrega do software dentro do prazo imposto.

Esta dificuldade encontrada com a metodologia *Scrum* poderia se tornar a base para a proposta de adaptações ao método, em especial quando o fator tempo estiver envolvido?

A proposta deste estudo de caso é verificar, na bibliografia existente, a sustentação para as ideias testadas, ou ainda a sua refutação devido a desvantagens que possam trazer em seu bojo e que tornariam alguma(s) das propostas, senão todas, menos atrativas.

Com base no acima exposto, o presente trabalho pretende analisar se as adaptações feitas à metodologia *Scrum* são aceitáveis, caso elas sejam pertinentes, tais adaptações seriam permitidas, ou melhor, seria uma proposta para criação de uma nova metodologia?

Este trabalho irá debater os prós e os contras das adaptações feitas e apresentar os resultados analisados para servir de base a futuras pesquisas sobre a possibilidade de adaptação da metodologia *Scrum* ou apresentação de uma nova proposta de metodologia ágil.

Metodologia

Foi feito levantamento de dados junto ao referencial teórico os quais foram confrontados com o conhecimento adquirido pelo grupo na disciplina de Engenharia de Software o que corrobora com a defesa dos métodos ágeis de desenvolvimento, frente aos métodos tradicionais, de forma a permitir uma maior interação do cliente com a fábrica de software possibilitando a adequação dos projetos à medida que os mesmos vão sendo desenvolvidos.

A literatura pesquisada e os conhecimentos adquiridos na graduação apontam para a importância da adaptação das metodologias existentes às necessidades do projeto, assim como os métodos ágeis foram a evolução dos métodos tradicionais, novas evoluções são bem vindas frente aos novos desafios que aparecem.

Resultados e discussões

No estudo de caso presente o desenvolvimento do software precisava ser entregue num prazo não usual para o trabalho daquela fábrica e alguns problemas foram levantados e enfrentados, os principais obstáculos e ajustes que são objeto do presente trabalho foram:

- O representante da área de negócio, ou cliente, é conhecido com *Product Owner* (ou dono do produto em uma tradução livre), também conhecido apenas pela abreviação como PO, ele deve ser capaz de passar para o time de desenvolvimento quais são os desejos do cliente, quais as regras que farão parte da aplicação, e o responsável pelo time *Scrum* pela coleta de tais regras é o Analista de Requisitos, este técnico deve ser capaz de entender o desejo do cliente, colocar em um documento de visão e levar para a aprovação do PO, e só depois passar para os desenvolvedores começarem a programação (Stherland e Schwaber, 2020).

A observância à metodologia *Scrum*, neste requisito, estava levando a inviabilidade da entrega do software no prazo necessário, a adaptação proposta, no caso em estudo, colocou o PO em contato diretamente com os desenvolvedores, permitindo que a construção da aplicação fosse imediata e todas as dúvidas fossem tiradas à medida que elas fossem surgindo, o Analista de Requisitos buscava PO para documentar todas as necessidades que ele estava passando diretamente aos desenvolvedores e assim fazer a documentação necessária, para registro e posteriormente faturamento da fábrica de software.

- A metodologia *Scrum* prevê uma reunião diária, chamada de *Daily Scrum*, que ocorre no início dos trabalhos onde, de maneira informal, todo o time fica em pé e discute por, no máximo 15 minutos, o que fez no dia anterior, o que fará no dia de hoje e se tem algum impedimento para ser removido para que possa fazer o seu trabalho (Stherland e Schwaber, 2020).

Mais uma vez, a metodologia que se mostra altamente eficiente em projetos “normais” começou a demostrar necessidade de ajustes neste caso em estudo, pois o desenvolvimento rápido de algumas funcionalidades deixava o desenvolver oscioso o restante do dia, ou mesmo um impedimento que atrasava a entrega de um desenvolvedor em específico ficava pendente da próxima reunião diária para que o *Scrum Master*, ou gerente da equipe, pudesse se envolver e trabalhar no sentido de remover o obstáculo. Também neste caso a adaptação foi aceita de forma bastante natural, as interações entre o gerente do time e os desenvolvedores ocorria a todo o momento, de forma que obstáculos detectados já eram removidos e funcionalidades entregues já podiam ser testadas no mesmo dia, pela equipe de testes, enquanto o desenvolvedor não ficava oscioso, pelo contrário, já se prontificava com novas metas e começava o desenvolvimento sem precisar esperar a próxima *Daily Scrum*.

Seguindo a metodologia *Scrum*, mas com as adaptações mencionadas, o software foi entregue dentro do prazo, não apernas o cliente ficou satisfeito com o resultado, mas o time todo

também ficou contente em poder atingir uma meta tão ousada, a cinergia criada acabou levando a uma maior integração no time que desenvolveu a aplicação.

Para buscarmos o entendimento sobre a possibilidade das adaptações elencadas acima se buscou na bibliografia existente a sustentação ou a refutação das observações, sendo possível separar os autores entre estas duas linhas.

Entre aqueles que entendem que a metodologia *Scrum* não deve sofrer alterações podemos apontar autores como Moraes, Borges e Okuyama (2013), que descrevem a importância da observância à metodologia da forma como ela é proposta, pois assim dará uma sustentação maior às empresas, principalmente quanto à documentação correta do projeto e que posteriormente permitirá uma manutenção adequada, no caso de projetos de software, além de permitir que a empresa busque o seu faturamento de uma forma mais fácil e transparente.

Na linha oposta encontramos autores como Ayed et. al (2014), que entendem que adaptações são necessárias para cada tipo de projeto, assim pode-se aproveitar o melhor da metodologia *Scrum* em qualquer projeto, bastando para tanto que adaptações sejam feitas, seria como abrir uma caixa de ferramentas e, caso a ferramenta encontrada não seja a mais adequada para o uso que se pretende, cria-se uma nova ferramenta, tomando como base as que já existem, e obtém-se a ferramenta ideal para o emprego no objetivo que se deseja alcançar.

Ao estudarmos os autores da metodologia Scrum, J. Stherland e K. Schwaber (2020), extraímos de seus trabalhos a questão de que alterações no *Scrum* descaracterizariam o *framework* e assim não estaríamos mais frente à metodologia por eles proposta, o *Scrum* deve ser seguido como uma receita de bolo, e uma alteração na receita resultariam em outro produto, mas não no produto chamado *Scrum*.

Bibliografia

- APOLLO 13. Direção: Ron Howard. Produção: Brian Grazer. Intérpretes: Tom Hanks; Ed Harris; Bill Paxton; Kevin Bacon; Gary Sinise; Kathleen Quinlan; Mary Kate Schellhardt; Emily Ann Lloyd; Miko Hughes; Max Elliott Slade; Jean Speegle Howard; David Andrews; Cris Ellis e outros. Roteiro: William Broyles, Jr. e Al Reinert. Universal Studios, 1995 (140 min).
- AYED, H. VANDEROSE, B. HABRA, N. **Supported Approach for Agile Methods Adaptation**: An Adoption Study. RCoSE, Hyderabad. India, 2014

- FREITAS, E. PRODANOV, C. **Metodologia do trabalho científico: métodos e técnicas de pesquisa**. 2ª edição. Universidade Feevale, 2013.
- MORAES, M. BORGES, K. OKUYAMA, F. **Autorregulação da aprendizagem em computação com apoio da metodologia Scrum**. Texto Livre: Linguagem e Tecnologia, [S.l.], v. 6, n. 2, p. 66-77, out. 2013.
  Disponível em: http://www.periodicos.letras.ufmg.br/index.php/textolivre/article/view/5066/7226.
  Acesso em: 25 mai. 2022. https://doi.org/10.17851/1983-3652.6.2.66-77
- SCHWABER, K. e SUTHERLAND, J. **O guia do Scrum**. O guia definitivo para o Scrum: as regras do jogo. 2020. Disponível em: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-2.0.pdf . Acessado em: 12 de maio 2022.
- SUTHERLAND, J. **Scrum a arte de fazer o dobro do trabalho na metade do tempo**. Editora LeYa. São Paulo, 2014.