

Adoção do método mínimo produto viável (MVP) em uma Fábrica de Software Acadêmica

Bruno D’Lucca Silva Carvalho, Marcella Canedo Tristão

Instituto de Ensino Superior – Centro Universitário de Anápolis - (UniEvangélica)
Caixa Postal 122 e 901 – Anápolis – GO – Brazil

{bruno.carvalho,marcella.tristao}@aluno.unievangelica.edu.br

Abstract. *The objective of this paper is to propose the adoption of the development of minimum viable products to the software projects of Fábrica de Tecnologias Turing and analyses the possibility of this adoption contribute to the projects improvement, elicitation and validation of functional and non-functional requirements, feedback gathering from final users and client, and functionalities prioritization, besides allowing that the developed software be in conformance with the client’s expectations and needs.*

Keywords: *minimum viable products, elicitation and validation, projects.*

Resumo. *O objetivo deste trabalho é propor a adoção do desenvolvimento de mínimos produtos viáveis para os projetos de software da Fábrica de Tecnologias Turing e analisa a possibilidade dessa adoção contribuir para a aprovação de projetos, elicitação e validação de requisitos funcionais e não funcionais, obtenção de feedback dos usuários finais e do cliente e priorização das funcionalidades, além de permitir que o software desenvolvido esteja em conformidade com as expectativas e necessidades do cliente.*

Palavras-chave: mínimos produtos viáveis, elicitação e validação, projetos.

1. Introdução

A utilização da metodologia ágil exige a compreensão dos princípios do Manifesto Ágil e a sua abordagem iterativa e incremental para o desenvolvimento de um produto. Estas ideias são conhecidas como “mentalidade ágil” mas poderia ser tratado como “pensar ágil”. A adoção de novas metodologias para complementar os métodos ágeis pode eventualmente exigir uma mudança organizacional dentro da empresa ou setor que irá eventualmente ter que adaptar-se ao novo método (SILVA, T. M. D. 2019).

A Fábrica de Tecnologias Turing (FTT), unidade dos cursos bacharelados em computação, é uma fábrica de *software* com sua estruturação dividida em três núcleos (Fábrica de *Software*, Núcleo de Capacitação e Núcleo de Pesquisa), seu principal objetivo é de formar alunos com uma visão mercadológica, isto é com um perfil humano, criativo e reflexivo, com experiência e conhecimentos necessários para o mercado de trabalho atual (UNIEVANGÉLICA, 2017). A Fábrica de Tecnologias Turing trabalha com a metodologia ágil de desenvolvimento híbrida composta pelo *Scrum*, possuindo foco no gerenciamento das equipes e de processos e o *OpenUP*, que aplica o desenvolvimento de *software* de modo iterativo e incremental ao processo.

Após a elicitação de requisitos o *Backlog*, que consiste em uma lista ordenada de tudo que deve ser desenvolvido no produto, deve ser priorizado pelo *Product Owner* a partir das necessidades do cliente, e é uma origem dos requisitos para qualquer mudança a ser feita no sistema.

No âmbito da FTT ao decorrer do desenvolvimento dos projetos notou-se que certos requisitos possuem alta volatilidade, e mesmo que estejam bem definidos, posteriormente à sua especificação sofrem refatorações devido à mudanças decorrentes novas necessidades descobertas em reuniões com o cliente, com o *product owner* ou com os orientadores.

Consoante a tese apresentada ao longo da introdução, este relatório visa propor a implementação do *Minimum Viable Product* (MVP) no processo iterativo e incremental da Fábrica de Tecnologias Turing, visando a melhoria do processo e a agilidade em obter um *feedback* do cliente.

2. Referencial Teórico

O Referencial teórico se dá início com a delimitação dos temas abordados, apresentando aspectos relacionados à obtenção de mínimos produtos viáveis para os projetos de *software*, assim como a metodologia ágil abordada e como a adoção da prática MVP impactaria no desenvolvimento dos projetos da FTT.

2.1. *Minimum Viable Product*

A sociedade contemporânea, dispõe de um conjunto comprovado de técnicas visando administrar grandes empresas, e conhecimento em melhores práticas para construir projetos reais.

O MVP é aquela versão do produto que permite uma volta completa do ciclo construir-medir-aprender, com o mínimo de esforço e o menor tempo de desenvolvimento. O produto mínimo viável carece de diversos recursos que podem se provar necessários mais tarde. No entanto, de certa forma, criar um MVP requer trabalho extra: devemos ser capazes de medir seu impacto. Por exemplo, não é adequado construir um protótipo que seja avaliado por engenheiros e designers apenas em função da sua qualidade interna. Também precisamos colocá-lo diante dos possíveis clientes para avaliar a reação deles. Pode até ser preciso tentar lhes vender o protótipo. (RIES, Eric, 2012, p.58).

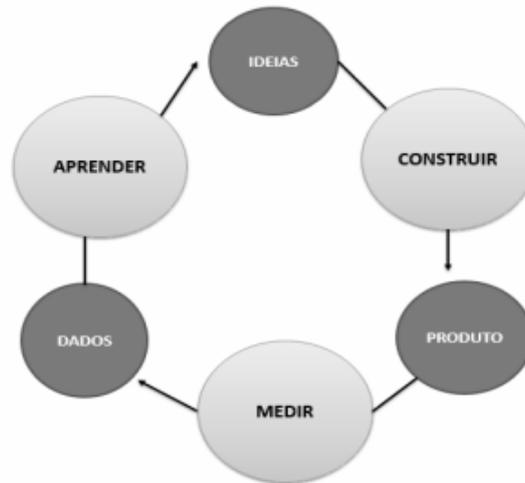


Figura 1. Ciclo de Feedback “Construir-Medir-Aprender”

Fonte: Adaptado de Ries (2012).

2.2. Scrum

O *Scrum* é um *framework* estrutural que é utilizado para o gerenciamento do desenvolvimento de produtos/sistemas complexos desde o início de 1990 com o manifesto ágil. O *Scrum* se trata de um *framework* no qual é possível empregar vários processos ou técnicas a fim de obter um resultado final. O *Scrum* deixa explicitamente a eficácia relativa das práticas de gerenciamento e de desenvolvimento de produtos, de modo em que seja flexível e que seja possível melhorá-los. (SCHWABER, Ken; SUTHERLAND, Jeff , 2013).

2.3. OpenUp

OpenUp é um processo de desenvolvimento de *software* minimamente suficiente, o que significa que apenas o que é necessário está incluído no processo. No entanto, OpenUp é completo no sentido de que pode ser utilizado como um processo inteiro para construir um sistema. É um processo ágil e aplica abordagens iterativas e incrementais com um ciclo de vida estruturado, valorizando a colaboração da equipe e os benefícios para as partes interessadas em relação a entregas e formalidades (ECLIPSE FOUNDATION, 2019).

2.4. eXtreme Programming

Extreme programming (XP) é uma metodologia ágil que enfatiza o trabalho em equipe. Gerentes, clientes e desenvolvedores são todos parceiros de igual importância em uma equipe colaborativa. O XP implementa um ambiente simples e efetivo, permitindo que as equipes sejam altamente produtivas, com foco em comunicação, simplicidade, *feedback*, respeito e coragem (WELLS, Don, 2013).

3. Abordagem Proposta / Estudo na FTT

3.1 Metodologia Híbrida da FTT

Na FTT é utilizada a metodologia ágil de desenvolvimento híbrida sendo composta pelo *Scrum*, possuindo foco no gerenciamento das equipes, e o *OpenUp* que possui o seu foco voltado para a parte da gerência dos processos, além de incluir algumas práticas do XP que são utilizadas pela equipe de desenvolvimento.

O Scrum no âmbito da FTT é o *framework* responsável por todo o processo de produção de sistemas garantindo a transparência, inspeção e adaptação entre os membros do projeto. Os projetos são afetados por restrições de tempo, custo, escopo, qualidade, recursos, capacidade de organização, dentre outras limitações (SBOK GUIDE, 2016).

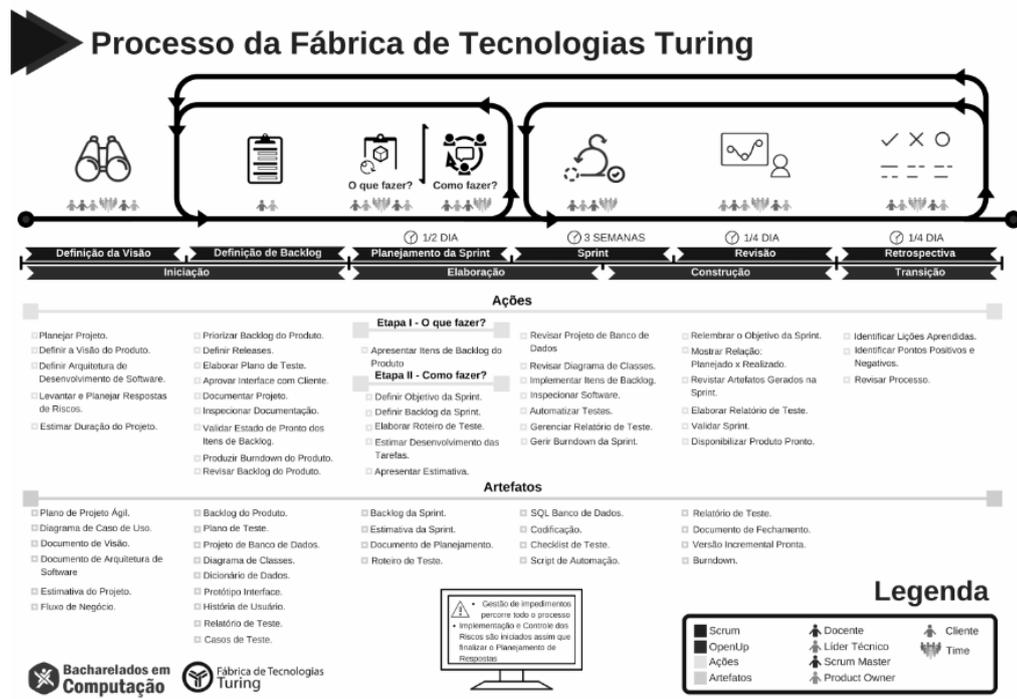


Figura 2. Processo da Fábrica de Tecnologias Turing

O OpenUP é utilizado na fábrica para definir os artefatos que devem ser produzidos, como documentos de especificação de requisitos, casos de teste, relatos de erros e correções, entre outros, e definir papéis dentro da equipe da FTT juntamente com o Scrum.

Assim, os papéis desempenhados dentro da FTT pelos acadêmicos integrantes são:

- **Product Owner (Scrum)** - O *product owner* (PO) é o responsável pelo gerenciamento do *backlog* e por garantir que todos compreendam claramente cada item nele expresso (SCHWABER, Ken; SUTHERLAND, Jeff, 2013);
- **Scrum Master (Scrum)** - O *scrum master* é o responsável por garantir que todos entendam e sigam os princípios e práticas do Scrum, além de auxiliar todos da equipe e remover os impedimentos (SCHWABER, Ken; SUTHERLAND, Jeff, 2013);

- *Developer* (OpenUp) - O *developer* (desenvolvedor) é o responsável por desenvolver funcionalidades do *software*, implementá-las e integrar os componentes que fazem parte da solução (ECLIPSE FOUNDATION, 2019);
- *Analyst* (OpenUp) - O *analyst* (analista) é o responsável por receber informações dos *stakeholders* e/ou do PO para entender o problema a ser resolvido e então modelar e documentar os requisitos provenientes destas informações (ECLIPSE FOUNDATION, 2019);
- *Tester* (OpenUp) - O *tester* é o responsável por identificar, definir, implementar e conduzir os testes necessários, bem como registrar e analisar os resultados do processo de teste (ECLIPSE FOUNDATION, 2019).

O conteúdo do *OpenUP* trata da organização do trabalho pessoal, das equipes e das partes interessadas do projeto. O *OpenUP* ajuda a equipe a concentrar adequadamente seus esforços através da iteração ciclo de vida, a fim de agregar valor incremental às partes interessadas de maneira previsível.

Das práticas do XP utilizadas na FTT, podemos destacar a padronização de código, que significa que todo código produzido deve obedecer a um padrão estabelecido, programação em pares que estabelece que todo código deve ser produzido por uma dupla de programadores, e a propriedade coletiva de código, que significa que todos têm acesso a todo código produzido por outras duplas.

3.2 Inclusão do MVP na metodologia da FTT

O MVP não deve necessariamente ser o menor produto imaginável e não deve ser perfeito, mas deve ser a maneira mais rápida de percorrer o ciclo construir-medir-aprender de *feedback* com o menor esforço possível. Ao contrário de testes de conceito ou protótipos, o MVP não serve apenas para responder a perguntas técnicas, mas para testar hipóteses fundamentais do negócio (RIES, Eric, 2012).

A proposta consiste em incluir na metodologia da FTT o desenvolvimento de produtos mínimos viáveis, com o objetivo de entregar um produto de valor para os *stakeholders* com o mínimo de tempo possível e assim obter o seu *feedback* com base na experiência de uso do MVP.

Assim como pode ser observado na figura a seguir, na primeira situação, a primeira entrega é apenas um pneu, uma parte do todo que não pode ser utilizada sozinha. Nas entregas posteriores, mais partes são incluídas, porém ainda é impossível utilizar o produto. Somente na última entrega o cliente poderá de fato utilizar o produto, e no caso de suas necessidades de negócio não serem atendidas pelo produto, o projeto sofrerá mudanças ou será cancelado (KNIBERG, Henrik, 2016).

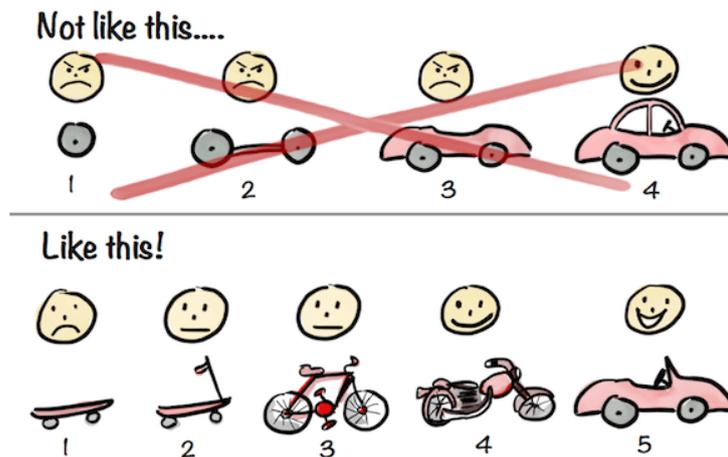


Figura 2. Ciclo do desenvolvimento MVP

Fonte: Henrik Kniberg.

Na segunda situação, o cliente, a cada entrega, recebe um produto que é possível de ser utilizado, porém que resolve atende às suas necessidades de forma muito simples e precária. No entanto, utilizando este produto, é possível que o cliente perceba quais são os pontos que estão de acordo com suas necessidades, quais pontos devem melhorar e o que está faltando e deve ser acrescentado ao produto (KNIBERG, Henrik, 2016).

Assim nas entregas seguintes, o produto é melhorado, mas sempre de modo que é um produto completo, mesmo que não atenda à todas as necessidades do cliente, pois o importante é a execução do ciclo construir-medir-aprender que consiste em construir o produto mínimo viável, o que no contexto da FTT é um *software* com funcionalidades básicas que atendem parcialmente o cliente, medir ou mensurar, fase em que os dados provenientes do *feedback* experiência de uso do cliente com o produto são analisados, e aprender, onde os novos requisitos e as possíveis mudanças solicitadas serão levantados e especificados para então se iniciar o ciclo novamente (RIES, Eric, 2012).

A metodologia híbrida utilizada na FTT é iterativa e incremental, o que pode facilitar a adoção do MVP juntamente com o ciclo construir-medir-aprender. A proposta consiste em no lugar de entregar funcionalidades individuais a cada *sprint*, o PO junto a toda a equipe estabeleçam quais requisitos irão compor o MVP, e então estimar o tempo necessário para a construção do mesmo e iniciar o seu desenvolvimento. Ao final do desenvolvimento, entregar o produto produzido ao cliente e solicitar o *feedback*. O PO e os analista serão os responsáveis por analisar os dados e modelar as novas funcionalidades e alterações.

4. Considerações Finais

Espera-se que com a inclusão do MVP na metodologia híbrida de desenvolvimento ágil da FTT possa contribuir na diminuição das mudanças e refatorações dos requisitos ao longo dos projetos de *software* desenvolvidos, na obtenção de *feedback* dos *stakeholders* e consequentemente na modelagem e elicitación de novos requisitos, de modo que a cada interação dos *stakeholders* com o produto produzido, possa-se receber o seu *feedback* com

base na sua experiência de uso, podendo assim serem reportadas quais funcionalidades estão prontas, quais precisam de ajustes e possíveis novas funcionalidades.

O desenvolvimento deste trabalho foi muito importante para a agregar conhecimento, pois com a observação do processo de desenvolvimento e os estudos realizados sobre a metodologia da FTT e sobre cada metodologia a ela incorporada, foi possível compreender os motivos da utilização de cada prática e método do Scrum, OpenUP e XP que compõe a metodologia da FTT e propor melhorias para a mesma.

5. Referências

ÁGIL, Desenvolvimento. Scrum. 2014. Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum/>>.

ECLIPSE FOUNDATION. Enabling Open Innovation & Collaboration, c2019. OpenUp Vision. Disponível em: <https://www.eclipse.org/epf/openup_component/openup_vision.php>. Acesso em: 14 de dez. de 2019.

Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., Abrahamsson, P.: **Software development in startup companies: A systematic mapping study**. Inf. Softw. Technol. 56(10), 1200–1218 (2014)

RIES, Eric; **A Startup Enxuta: Como os Empreendedores Atuais Utilizam a Invocação Contínua para Criar Empresas Extremamente Bem-sucedidas**. Edição. São Paulo: Lua de Papel, 2012. p. 1-210.

Scrumstudy (2016), “**A guide to the Scrum body of knowledge**” (SBOK GUIDE): Vmedu.

SCHWABER, Ken; SUTHERLAND, Jeff. **Guia do Scrum: Um guia definitivo para o Scrum: As regras do jogo**. Disponível em: <www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>. Acesso em: 2 dez. 2019.

SILVA, T. M. D. **Aplicação de métodos ágeis no desenvolvimento de software automotivo** : Conciliação ou substituição ao ciclo-v. Disponível em: <<https://www15.fgv.br/network/tcchandler.axd?tcid=8988>>. Acesso em: 12 dez. 2019.

UniEvangélica, C. U.(2017): **Engenharia de Software: Diferencias**, Disponível em:<<http://www.unievangelica.edu.br/curso.engenharia.software/diferenciais>>.

WELLS, Don. Extreme Programming: A Gentle Introduction, 2013. Página Inicial. Disponível em: <<http://www.extremeprogramming.org>>. Acesso em: 15 de dez. de 2019.