

REACT-M: O Relato de um Estudo de Caso de Aplicação de um Método Ágil para Gerência de Requisitos de Software

Bleno W. F. V. da Silva¹, Sandro R. B. Oliveira¹

¹ Programa de Pós-Graduação em Ciência da Computação, Faculdade de Computação - Instituto de Ciências Exatas e Naturais - Universidade Federal do Pará, Brasil
blenofvale@gmail.com, srbo@ufpa.br

Abstract. This study presents the report about a case study on the application of REACT-M (Requirements Evolution in Agile Context - Management), an agile method to support software requirements management, in a real software development environment. This case study assessed the strengths, weaknesses, opportunities, and threats related to the assets that make up REACT-M, namely: work products, life cycle, roles, and ceremonies. About the results, it was observed that REACT-M was effective, simple to use, strongly collaborative, suitable, user centered and goal oriented. Thus, achieved its main purpose to evolve the requirements of a software product in iteratively, efficiently and under of the agile mindset. The results contribute to the software industry by providing empirical data about the use of a agile method to software requirements management, which can serve as a reference for organizations seeking the adoption of agile methods related to the requirements area, as well as to provide the scientific community with a better empirical understanding of the relationship between requirements and agile methods.

Resumo. Este estudo apresenta o relato de um estudo de caso sobre a aplicação do REACT-M (*Requirements Evolution in Agile Context - Management*), um método ágil para a gerência de requisitos de software, em um ambiente real de desenvolvimento de software. Este estudo de caso avaliou as forças, as fraquezas, as oportunidades e as ameaças relacionadas aos ativos que compõem o REACT-M, a saber: artefatos, ciclo de vida, papéis e cerimônias. Dentre os resultados, observou-se que o REACT-M foi efetivo, simples de usar, fortemente colaborativo, *suitable*, centrado no usuário e orientado a metas. Assim, atendendo ao seu principal objetivo que é gerenciar a evolução dos requisitos de um produto de software de forma iterativa, eficiente e sob o guarda-chuva do *mindset* ágil. Os resultados contribuem à indústria de software fornecendo um relato de experiência sobre o uso de um método ágil para a gerência de requisitos de software, o qual pode servir de referência para organizações que buscam a adoção de métodos ágeis relacionados à área de requisitos, assim como, prover à comunidade científica um melhor entendimento empírico do relacionamento entre requisitos e métodos ágeis.

1 Introdução

Os métodos ágeis de desenvolvimento de software vêm ganhando crescente popularidade desde o início da década de 2000. Este fato fundamenta-se pelos diversos benefícios que os mesmos proporcionam à indústria de software, como melhor gestão de mudanças, aumento na produtividade da equipe, melhoria na qualidade de software e maior ênfase na comunicação com o cliente [19]. Além disso, estes métodos surgiram como uma alternativa para minimizar alguns dos problemas

das abordagens tradicionais, enfrentados por um mercado que exige cada vez mais rapidez na entrega de produtos de qualidade [9]. Alguns destes problemas estão relacionados aos requisitos, por exemplo pouco envolvimento dos clientes, alta mudança nos requisitos, requisitos mal especificados, dentre outros [21]. Entretanto, segundo Inayat *et al.* [4], Dyba e Dingsoyr [3] e Parsons *et al.* [7], ainda são escassas as evidências empíricas sobre as metodologias ágeis que estão sendo utilizadas em ambientes reais de desenvolvimento de software.

Dentre as áreas de conhecimento da engenharia de software, a engenharia de requisitos possui extrema importância no ciclo de vida de projetos de software [8, 16]. Segundo o *Chaos Report* publicado por Standish Group [17], cerca de 21% dos principais fatores de falhas em projetos de software estão relacionados com os requisitos, que possuem impacto direto na qualidade do produto final. Já em outro estudo realizado por Schwaber [11], o qual analisou dados de mais de 100 projetos de desenvolvimento de software, o autor notou que 35% dos requisitos mudam, 65% das funcionalidades descritas pelos requisitos nunca ou raramente são utilizadas e cerca de 50% do tempo despendido da equipe é gasto com requisitos, arquitetura e especificação do produto. Com base nesta afirmativa, diversos modelos de qualidade para processos de software como o MR-MPS-SW (Modelo de Referência MPS para Software) [18] e o CMMI-DEV (*Capability Maturity Model Integration for Development*) [2] adotam guias para o desenvolvimento e gerência de requisitos.

Neste cenário, vem crescendo consideravelmente o número de abordagens ágeis para apoio ao desenvolvimento de produtos de software, surgindo diversos métodos, técnicas e práticas alinhadas aos princípios e valores do manifesto ágil que buscam resolver ou minimizar alguns dos problemas ou lacunas ainda existentes nos processos usados na indústria de software [5]. Neste contexto, a engenharia de requisitos ágil vem ganhando destaque, por sugerir o uso de práticas, princípios e valores dos métodos ágeis em suas atividades. Esta abordagem permite desenvolver e gerenciar os requisitos de forma mais ágil, utilizando documentação mais simples e concisa, eliminando alguns artefatos e documentos de controle que com frequência são identificados nas metodologias tradicionais [10]. Logo, torna-se significativo e justificável realizar uma pesquisa científica a fim de relacionar a área de Engenharia de Requisitos (ER) com os métodos ágeis.

Além disso, é notável o aumento da preocupação com a qualidade do produto por parte da indústria de software, buscando melhor atender a necessidade dos usuários e também tornando este ponto um diferencial no mercado para as organizações. Para isto, tem-se investido bastante na melhoria do processo de desenvolvimento de software [6]. Desta forma, como já mencionado, diversos modelos de qualidade para processos de software como o MR-MPS-SW e o CMMI-DEV surgem com o objetivo de orientar o caminho da melhoria. Os modelos citados possuem preocupação com as atividades de Engenharia de Requisitos e fornecem boas práticas que devem ser adotadas para o Desenvolvimento de Requisitos (DRE), que abrange as atividades de elicitação, análise, especificação e validação de requisitos, e também para a Gerência de Requisitos (GRE), que envolve atividades de gerenciar o *Backlog*, rastreabilidade, gerência de mudança, revisão de requisitos [2, 18, 21]. É importante ressaltar que o presente estudo está inserido no contexto do processo de Gerência de Requisitos, por ser uma extensão do estudo conduzido por Santos e Oliveira [10], que possuía foco nas atividades de desenvolvimento de requisitos. Assim, este trabalho pretende tratar de uma lacuna do trabalho anterior, fornecendo um controle e evolução dos requisitos em ambiente ágil.

Neste cenário, Silva *et al.* [15] propuseram um método ágil específico para a área da ER denominado REACT-M (*Requirements Evolution in Agile Context - Management*), um método que foi concebido a partir de outros métodos ágeis. É importante frisar que a proposta do REACT-M é de apenas gerenciar os requisitos de software, ou seja, compreende todas as atividades pertencentes ao processo de GRE, uma subárea da ER responsável pelo controle e evolução dos requisitos ao longo do ciclo de vida do projeto até se obter um produto funcional de valor e que atenda às

reais expectativas dos *stakeholders* [18]. Em suma, este processo lida com o controle de mudanças, o controle de versão, a revisão e o rastreamento dos requisitos de um produto de software [21].

Portanto, o presente estudo tem como objetivo apresentar os resultados de um estudo de caso que aplicou e avaliou o método ágil REACT-M em um ambiente real de desenvolvimento de software.

O restante deste estudo está estruturado da seguinte forma: a Seção 2 discute sobre alguns trabalhos relacionados; a Seção 3 apresenta sucintamente o REACT-M; a Seção 4 relata o Estudo de Caso; a Seção 5 apresenta e discute as limitações do estudo; e a Seção 6 apresenta as conclusões e trabalhos futuros.

2 Trabalhos Relacionados

Alguns estudos conduziram pesquisas empíricas a fim de avaliar novas propostas de abordagens ágeis para a área de ER. Dentre estes, destaca-se o estudo realizado por Medeiros [5] que conduz uma revisão sistemática da literatura com o intuito de identificar como a engenharia de requisitos está sendo tratada em projetos ágeis de desenvolvimento de software. O trabalho identificou diversas abordagens ágeis para apoio às atividades de engenharia de requisitos, como por exemplo as histórias de usuário e os cenários de aceitação. No entanto, o estudo apenas discorre sobre técnicas de elicitação e especificação dos requisitos, deixando de lado outras atividades importantes, como as de gerência de requisitos.

O estudo de Inayat *et al.* [4] desenvolveu uma revisão sistemática da literatura a fim de identificar práticas ágeis voltadas à engenharia de requisitos e avaliar quais os desafios estão relacionados a estas práticas, como por exemplo a negligência de requisitos não-funcionais, a documentação insuficiente e a pouca disponibilidade do cliente. A partir da RSL desenvolvida, os pesquisadores propuseram um método que contempla algumas das principais atividades de DRE, como elicitação, especificação e validação.

O estudo conduzido por Sehrish e Shah [13] realizou uma revisão sistemática da literatura com o objetivo de apresentar as limitações encontradas pela engenharia de requisitos em ambientes ágeis e interpretar quais são os problemas e desafios que uma pessoa ágil enfrenta na implementação de práticas ágeis. Os resultados contribuem para um melhor entendimento sobre como resolver alguns problemas particulares de requisitos no processo de desenvolvimento de software com métodos ágeis.

Embora os três estudos tenham como um dos objetivos a identificação de práticas ágeis para a engenharia de requisitos, este trabalho difere-se por também abranger métodos e técnicas ágeis para apoio à gerência de requisitos em sua revisão sistemática de literatura. Além disso, os estudos relacionados não buscam definir um processo que cubra todas as principais atividades de engenharia de requisitos, deixando de lado diversas atividades importantes quando se trabalha com requisitos em projetos de software, principalmente quando se trata de atividades direcionadas para a gerência de requisitos, ou seja, as abordagens ágeis destes estudos não contemplaram e avaliaram o uso de todas as atividades de GRE.

3 REACT-M: Um Método Ágil para Gerência de Requisitos de Software

O REACT-M é um método ágil com o objetivo de gerenciar os requisitos de um produto de software de forma colaborativa, eficiente, iterativa e sob o guarda-chuva do *mindset* ágil. Este método foi concebido a partir de três estratégias de pesquisa pensadas para que sua fundamentação fosse considerada fortemente empírica, a saber:

a) Análise de Abordagens Ágeis encontradas na literatura por meio de um Mapeamento Sistemático da Literatura (MSL); b) Análise de Práticas da *Agile-RE* (*Agile Requirements Engineering*); e c) Análise das boas práticas constantes nos modelos de qualidade CMMI-DEV e MR-MPS-SW, especificamente, em seus respectivos processos REQM (*Requirements Management*) e GRE.

Para saber mais sobre o REACT-M, dois artigos foram publicados em dois eventos internacionais: o primeiro artigo de Silva e Oliveira [14] disserta sobre o MSL conduzido; enquanto que o artigo de Silva *et al.* [15] disserta sobre a criação, o funcionamento e as regras do REACT-M. Mais detalhes do REACT-M em: https://drive.google.com/file/d/10VSJdkVaB2a4j_wHqaQ9IxaKEChOOgG/view?usp=sharing.

4 Estudo de Caso de Aplicação do REACT-M

Este estudo de caso foi realizado em uma unidade de desenvolvimento de software dentro da Universidade Federal do Pará (UFPA). Esta unidade foi escolhida por: ter interesse no estudo proposto; ter um ambiente favorável e acessível à execução do estudo de caso; e possuir projetos de software que possibilitariam e seriam beneficiados com a execução do REACT-M.

Um estudo de caso foi escolhido como método de pesquisa por se tratar de uma abordagem qualitativa observacional, onde o pesquisador interage de forma semiformal com os sujeitos da pesquisa por meio de entrevistas e conversas programadas, a fim de levantar as avaliações ou opiniões destes sujeitos sobre um determinado fenômeno, solução, método, processo ou prática inserido no ambiente destes sujeitos [20]. Este estudo de caso foi organizado em três fases distintas, conforme apresentado na Figura 1.

Adicionalmente, a estratégia metodológica deste estudo de caso é apresentada no Quadro 1. Este quadro foi criado a partir das recomendações de Wainer [20] e de Silva e Menezes [12], os quais dissertam sobre as características de um estudo de caso. Cada fase e suas respectivas etapas serão melhor detalhadas nas próximas subseções.

4.1 Fase de Planejamento

Em primeiro lugar, a definição do tipo de projeto e da Equipe foi necessária. Para isto, ambos deveriam atender a duas premissas: a) estar inserido em um contexto ágil de desenvolvimento de software, ou seja, a Equipe deveria fazer uso dos métodos ágeis para desenvolvimento de seus produtos de software; e, b) conduzir um projeto de software onde o produto ainda não esteja bem definido e ainda precisa ser descoberto. Este ambiente seria um dos melhores ambientes para avaliar o REACT-M, uma vez que o REACT-M caracteriza-se como um método ágil para a evolução e o controle dos requisitos de um produto.

Logo, foi definida uma Equipe composta por 5 **Desenvolvedores** (é a equipe de desenvolvimento responsável por evoluir os requisitos do produto ao longo do projeto), 1 **Cliente** (para quem o produto de software está sendo desenvolvido, tendo como objetivo de tomar decisões sobre os requisitos ao longo do projeto), 1 **Especialista de Domínio** (é o representante do cliente, que possui uma grande *expertise* em determinadas áreas de domínio dentro do negócio do cliente, tendo como objetivo de ajudar o cliente e os desenvolvedores a desenvolver os requisitos de um produto que realmente atenda às necessidades do negócio) e 1 **Facilitador do REACT-M** (é uma espécie de *coach* com grande *expertise* sobre o REACT-M, responsável por guiar todos os envolvidos com interesse na evolução dos requisitos a aplicar o REACT-M da melhor forma possível em cada contexto de projeto de

software. Aqui teve-se a participação dos autores do método.). Esta Equipe possuía a responsabilidade de conduzir um projeto de desenvolvimento de um software destinado a ensinar aos desenvolvedores programação *Front-end*, assim torná-los mais capacitados para a execução de outros projetos conduzidos dentro da unidade de desenvolvimento de software da UFPA, bem como, este produto ainda estava cercado de incertezas e o próprio cliente ainda não possuía uma ideia bem clara sobre o mesmo. O sistema trata-se de um portal para registro das produções da equipe do projeto SPIDER – *Software Process Improvement: DEvelopment and REsearch*. Este cliente era um pesquisador da UFPA, aluno de doutorado do PPGCC da UFPA.

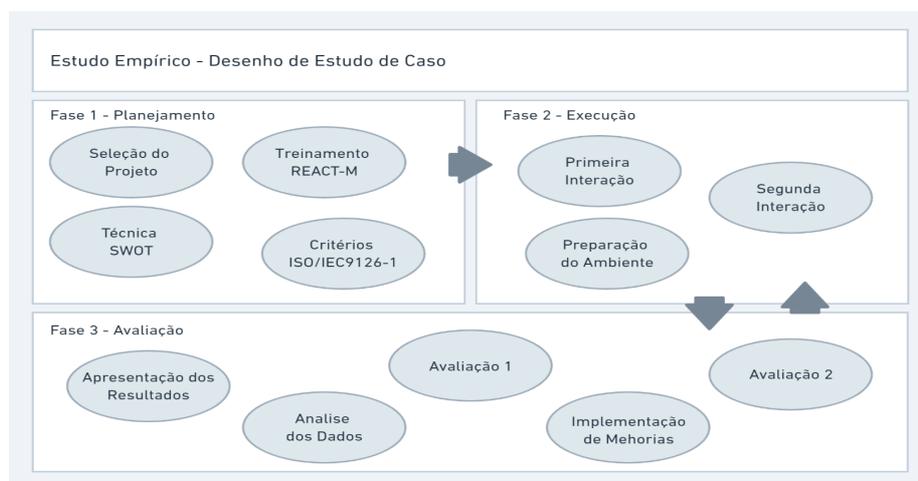


Fig. 1. Desenho do Estudo de Caso.

Quadro 1. Quadro metodológico do Estudo de Caso.

Tipo de Pesquisa	Exploratória
Procedimento de pesquisa	Estudo de caso
Natureza dos dados	Qualitativa
Tipo de registro dos dados	- Anotações de campo - Transcrição dos <i>Brainstormings</i>
Instrumento para coleta e análise dos dados	- Análise documental - <i>Brainstorming</i> - Aplicação da técnica de Análise SWOT, a fim de identificar forças, fraquezas, oportunidades de melhoria e ameaças quanto à aplicabilidade do REACT-M.

Este estudo de caso compreendeu duas iterações de desenvolvimento com duração de 14 dias úteis cada para a aplicação do REACT-M. Ao final de cada iteração, uma avaliação foi aplicada junto à Equipe. Para esta avaliação, a técnica de análise SWOT (*Strengths, Weaknesses, Opportunities e Threats*) foi escolhida e adaptada para o contexto desta pesquisa. Deste modo, objetivando avaliar os ativos que compõem o REACT-M, lembrando: ciclo de vida, artefatos, papéis e cerimônias; a partir das perspectivas de forças, fraquezas, oportunidades de melhoria e ameaças.

Outrossim, critérios de avaliação foram definidos com o intuito de ajudar a Equipe ao longo da fase de avaliação do REACT-M. Optou-se por utilizar alguns critérios adaptados da norma ISO/IEC 9126-1 [1], devido esta norma ser uma referência quanto à avaliação de qualidade de produto de software, a saber: Eficiência, Simplicidade, Utilidade, Aplicabilidade, Confiabilidade e Colaboração. Deste modo, ao final da avaliação os participantes foram convidados a fornecer uma nota geral sobre o uso do REACT-M, conforme cada critério de avaliação. A escala da nota

caracteriza-se da seguinte forma: a) Nota 0: para este critério o REACT-M não atendeu aos seus objetivos; b) Nota 5: para este critério o REACT-M atendeu parcialmente aos seus objetivos; e c) Nota 10: para este critério o REACT-M atendeu totalmente aos seus objetivos.

Por fim, um pequeno treinamento junto à Equipe foi conduzido, com a finalidade de ambientar os participantes do estudo de caso sobre as regras e o funcionamento do REACT-M. Um Guia Prático sobre o REACT-M foi entregue a cada participante para um melhor entendimento sobre o funcionamento do método.

4.2 Fase de Execução

Durante a execução do REACT-M, o Facilitador e o Cliente participaram de todas as cerimônias do método junto com o restante da Equipe, o que proporcionou um ambiente fortemente colaborativo e participativo, como é tipicamente sugerido no uso de métodos ágeis.

O primeiro passo nesta fase foi preparar o ambiente com os materiais e ferramentas que a Equipe iria utilizar para a aplicação do REACT-M. A Equipe optou por utilizar recursos como *post-it*, parede e folhas de papel A4 para criar e manter os artefatos durante o momento de ideação do produto. No entanto, a Equipe também definiu a utilização da ferramenta *Whimsical* para o gerenciamento do *Backlog* e dos artefatos correlatos, porque a ferramenta simula a utilização de *post-its* e paredes, porém sem as limitações físicas.

Primeira Iteração de Desenvolvimento: inicialmente, foi realizada uma cerimônia para o planejamento do produto que seria desenvolvido, onde nesta reunião todos os participantes do estudo de caso estavam presentes. O Cliente também possuía muitas dúvidas sobre o produto que realmente iria satisfazer os seus usuários finais. Para auxiliar nessa etapa de descoberta, o Facilitador primeiramente aplicou a técnica de *Personas* conforme Figura 2, onde todos os participantes interagiram a fim de mapear os objetivos dos fornecedores de requisitos do produto, que neste caso era um grupo formado pelos usuários-chave e o próprio cliente. Foi preciso o Facilitador aplicar abordagens ágeis para o desenvolvimento dos requisitos do produto, para isso foi usada uma adaptação de *Canvas*, que permitiu a equipe extrair das *Personas* quais *Features* o produto precisaria possuir, possibilitando uma visão de alto nível do produto como um todo. Para cada *Feature* identificada, foi utilizada a técnica de *Journey* que permitiu a equipe ter uma visão a partir de fluxo da interação da *Persona* com a *Feature* implementada no produto. Esta visão permitiu a extração dos PBI (*Product Backlog Items*) que deveriam ser implementados no produto para entregar a *Feature*. Os PBI foram escritos no formato de *User Stories*, que facilita o entendimento do Cliente sobre o que seria desenvolvido. O conjunto desses PBI formou o *Backlog* do produto que foi priorizado pelo cliente com a técnica de *User Story Mapping*.

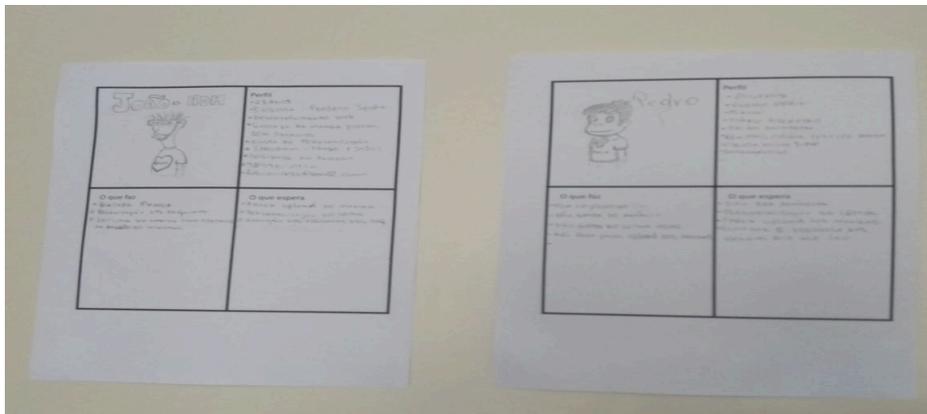


Fig. 2. *Personas* mapeadas do Produto.

As abordagens de *Journeys* e *User Stories* são sugeridas pelo método REACT, pois o método REACT-M é especificamente voltado para a gerência de requisitos.

Terminada esta etapa, os PBI foram migrados para o *Domain Expert Kanban* criado virtualmente na ferramenta *Whimsical*, conforme Figura 3. Todos os PBI foram colocados na lista *Product Backlog* do *Domain Expert Kanban*, com exceção dos PBI de maior prioridade e que foram definidos pelo cliente para a primeira entrega. Estes foram movidos para a lista *Sprint Backlog*, dando visão para todos do que deve ser desenvolvido na primeira iteração. Para cada PBI foi atribuído um código identificador único e cada PBI, caso o PBI possua dependência com outro, era colocado no cartão o código destes PBI gerando a rastreabilidade horizontal entre eles.

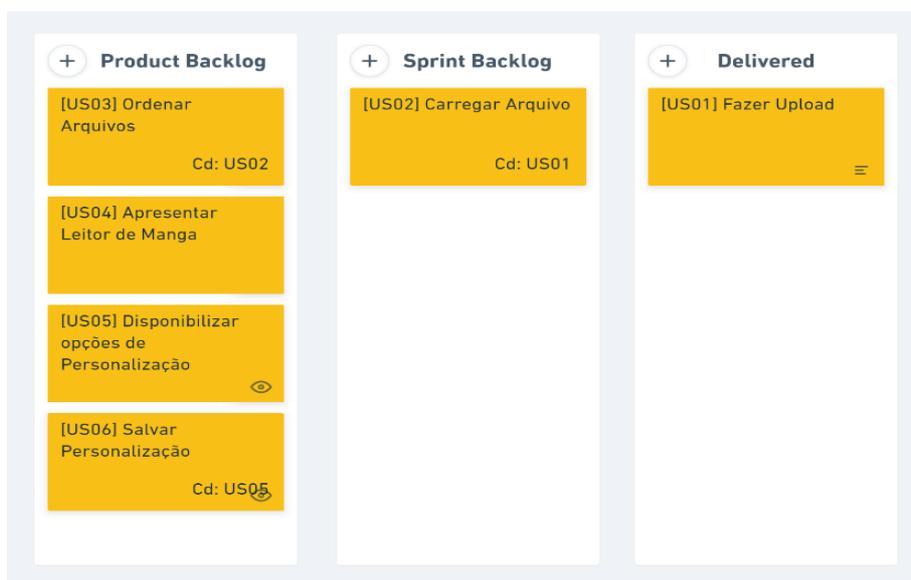


Fig. 3. *Domain Expert Kanban*.

Para facilitar o fluxo de trabalho da Equipe dentro da iteração, como proposto no REACT-M, foi criado o *Team Kanban*, conforme a Figura 4, que tem foco no gerenciamento dos PBI que estão sendo desenvolvidos dentro da iteração corrente. Com o entendimento do que precisa ser desenvolvido na primeira entrega, os participantes puderam voltar suas atenções para estes PBI, e com a ajuda do Cliente e do Especialista do Domínio foram desenvolvidos os Critérios de Aceite destes PBI utilizando os conceitos de *Definition of Ready*, que forneceu detalhes do negócio importantes para a implementação dos PBI.

Os critérios de aceite foram também criados na ferramenta *Whimsical*, onde para cada critério de aceite foi atribuído um código de identificação único, e também era colocado o código dos PBI que o critério de aceite pertencia, gerando uma rastreabilidade vertical entre os artefatos. A Equipe ficou encarregada de validar as informações dos PBI utilizando os critérios INVEST (características que os requisitos devem ter para serem validados: *Independent, Negotiable, Valuable, Estimable, Small, Testable*), conforme a Figura 5, a fim de verificar se estavam adequados para serem desenvolvidos dentro da iteração.

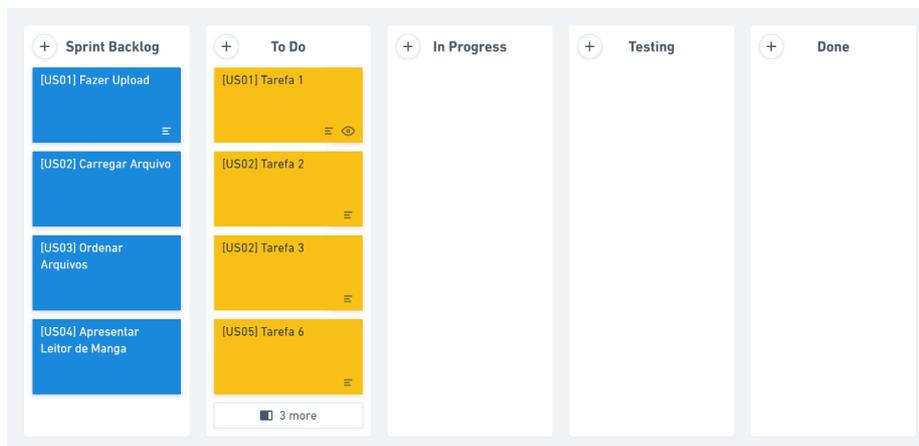


Fig. 4. Team Kanban.



Fig. 5. Critérios de Aceite.

Após a validação do PBI, a Equipe possuía informações suficientes para estimar o esforço que levaria para implementar os PBI. Para realizar a estimativa foi utilizada a técnica de *Planning Poker*, onde todos os membros participaram. Como resultado, chegou-se a um valor estimado de esforço para cada PBI e também a definição das tarefas técnicas que deviam ser implementadas. As Tarefas definidas foram criadas na lista *To Do* do *Team Kanban*, e para cada uma delas foi atribuído um código identificador único, também foi incluído o código identificador do PBI que a derivou, gerando rastreabilidade entre os artefatos de trabalho. Concluindo, foi estipulado, de acordo com a soma de esforço dos PBI, o prazo de 14 dias para o término da iteração.

Durante a iteração, a Equipe utilizou o *Team Kanban* para se auto gerenciar, sempre pedindo ajuda do Facilitador caso precisasse. Destaco que para um PBI ser dado como *Done* no *Team Kanban*, eram realizados testes no incremento gerado. O incremento precisava ser validado de acordo com os critérios INVEST. Caso fosse identificada alguma inconsistência, era aberto um defeito e inserido na lista *To Do* para ser corrigido.

Ao fim da iteração de desenvolvimento o cenário era de muitas incertezas e descobrimentos. Contudo, ao fim desta iteração todos sabiam quais requisitos eram os mais prioritários para serem desenvolvidos, assim como já possuíam especificações

suficientes e enxutas para iniciarem o desenvolvimento do produto, concomitantemente, com a execução do REACT-M.

Segunda Iteração de Desenvolvimento: antes de iniciar a segunda iteração de desenvolvimento, uma avaliação do REACT-M foi realizada junto à Equipe. Após isto, um conjunto de melhorias no REACT-M foi concebido para, então, executar a segunda iteração de desenvolvimento já contemplando estas melhorias. Estas melhorias podem ser visualizadas no Quadro 6.

A segunda iteração iniciou com uma solicitação de mudança da parte do cliente. Para realizar a mudança no escopo foram utilizadas as medidas propostas no REACT-M, que consistem em entender a mudança solicitada e basicamente executar cada cerimônia do método novamente para incluir a mudança no *Backlog*. Nesta segunda iteração, em síntese, a Equipe já detinha um entendimento e detalhamento mais significativo sobre os requisitos do produto, bem como um bom domínio sobre a execução do REACT-M, permitindo assim a execução das cerimônias do REACT-M de maneira mais eficiente e ágil. Para controlar as mudanças do projeto a Equipe utilizou o formulário proposto pelo REACT-M, conforme a Figura 6.

Mudanças						
N°	Data da Solicitação	Responsável	Descrição	PBIs		Esforço
				Impacto	Novas	
RQM01	26/10/2019	João	Incluir opção de pesquisa no Header do sistema.	US06	US07; US08	8

Fig. 6. Formulário para controle de mudanças.

Neste sentido, a segunda iteração corroborou com um dos principais objetivos do REACT-M: o controle de mudanças dos requisitos do produto conforme orientação de uso pelo método.

4.3 Fase de Avaliação

Ao final de cada iteração de desenvolvimento uma avaliação foi realizada junto à Equipe a fim de obter suas considerações e observações a respeito da aplicabilidade do REACT-M, ou melhor, esta avaliação permitiu a identificação das forças, das fraquezas, das oportunidades de melhoria e das ameaças relacionados aos ativos que compõem o REACT-M, a saber: ciclo de vida, artefatos, papéis e cerimônias.

Para isto, três instrumentos foram utilizados para a coleta e a análise dos dados durante a avaliação: a) as avaliações foram realizadas no formato de um *Brainstorming*, pelo qual o pesquisador avaliador deste estudo de caso conduziu uma sessão em grupo com os participantes deste estudo de caso com o objetivo de levantar suas considerações e observações sobre os ativos do REACT-M; b) a técnica de Análise SWOT foi aplicada por meio do Formulário de Avaliação SWOT, conforme visualizado na Figura 7; e c) a Equipe fez uma análise documental, ou seja, analisou todos os artefatos gerados ao longo do estudo de caso para fundamentar suas considerações e observações.

Desta forma, as transcrições dos *Brainstormings* e as anotações de campo foram registradas, analisadas e compiladas para gerar os resultados empíricos deste estudo de caso. Estes resultados podem ser conferidos na próxima seção.

Data: Iteração avaliada: Participantes:

Itens para avaliar: Ciclo de Vida Artefatos Papéis Cerimônias

Crítérios	S (<i>Strengths</i>)	W (<i>Weaknesses</i>)	O (<i>Opportunities</i>)	T (<i>Threads</i>)
Eficiência				
Simplicidade				
Utilidade				
Aplicabilidade				
Confiabilidade				
Colaboração da Equipe				

Fig. 7. Formulário de avaliação SWOT.

4.4 Sumarização dos Resultados do Estudo de Caso

Os resultados empíricos da avaliação do REACT-M estão sumarizados no Quadro 2, Quadro 3, Quadro 4 e Quadro 5, organizados sob as perspectivas do SWOT e dos critérios de avaliação definidos na Fase de Planejamento deste estudo de caso.

Quadro 2. Forças do REACT-M.

Crítérios	S (<i>Strengths</i>)
Eficiência	<p>Artefato: Equipe julgou que os artefatos ajudaram muito no desenvolvimento.</p> <p>Artefato: Equipe julgou que o artefato <i>Persona</i> foi um ótimo ponto inicial para direcionar as ideias do cliente a atingir o objetivo que era gerar o <i>Backlog</i> do Produto.</p> <p>Artefato: Equipe julgou que o artefato <i>Team Kanban</i> ajudou a gerenciar as tarefas que precisavam ser feitas na iteração.</p> <p>Papeis: Equipe julgou relevante a divisão dos papéis e atribuição das responsabilidades, de modo que qualquer problema se sabia a quem consultar.</p>
Simplicidade	<p>Papel: Equipe julgou simples as responsabilidades de cada papel.</p> <p>Artefato: Equipe julgou simples o uso dos artefatos <i>Domain Expert Kanban</i> e <i>Team Kanban</i></p> <p>Artefato: Equipe julgou simples o uso dos critérios INVEST para validar os itens do <i>Backlog</i>.</p>
Utilidade	<p>Artefato: A rastreabilidade foi bastante útil para ligar os artefatos, o que facilitou os desenvolvedores entenderem o que precisavam desenvolver.</p> <p>Ciclo de vida: Equipe achou interessante a ideia de dividir o que vai ser desenvolvido por ciclos, onde o que é prioridade para o cliente é implementado primeiro, mas levando também em consideração o esforço estimado pela equipe.</p>
Aplicabilidade	Não relatado pelos participantes.
Confiabilidade	Ciclo de vida: Todas as etapas e artefatos tem ligação entre si, não há lacunas.
Colaboração em Equipe	REACT-M: A participação de toda a equipe foi muito significativa em todas as cerimônias, sempre discutindo e compartilhando conhecimento sobre o produto.

Quadro 3. Fraquezas do REACT-M.

Crítérios	W (Weaknesses)
Eficiência	Cerimônias: Realizar a cerimônia de <i>Inception</i> toda em apenas um encontro
Simplicidade	Artefato: Equipe teve dificuldade de utilizar a ferramenta proposta para o gerenciamento do <i>Team Kanban</i> , onde se sugeriu a utilização de outra ferramenta como o Trello.
Utilidade	Artefato: Equipe julgou o uso dos critérios INVEST não tão relevante, tendo em vista que a Equipe participou de todo processo de levantamento e geração do <i>Backlog</i> , onde isto já indicava que estavam em conformidade com o <i>Backlog</i> .
Aplicabilidade	Artefato: Controlar as mudanças em um formulário separado não foi considerado interessante pela equipe.
Confiabilidade	Artefato: Equipe considerou realizar a rastreabilidade manual um ponto fraco, pois existem ferramentas que permitem gerar quase automática.
Colaboração em Equipe	Não relatado pelos participantes.

Quadro 4. Oportunidades do REACT-M.

Crítérios	O (Opportunities)
Eficiência	Artefato: Tornar a validação do <i>Backlog</i> tarefa implícita no caso de toda equipe participar do levantamento do <i>Backlog</i> .
Simplicidade	Artefato: Equipe considerou que os critérios de aceite poderiam ficar juntos com o item de <i>Backlog</i> , como uma descrição do item, no caso da utilização de ferramenta.
Utilidade	Não relatado pelos participantes.
Aplicabilidade	Não relatado pelos participantes.
Confiabilidade	Não relatado pelos participantes.
Colaboração em Equipe	Não relatado pelos participantes.

Quadro 5. Ameaças do REACT-M.

Crítérios	T (Threats)
Eficiência	Não relatado pelos participantes.
Simplicidade	Não relatado pelos participantes.
Utilidade	Não relatado pelos participantes.
Aplicabilidade	Artefato: Aplicar os critérios INVEST para o teste do incremento ficou um pouco difícil para a equipe entender.
Confiabilidade	Não relatado pelos participantes.
Colaboração em Equipe	Papeis: Em alguns momentos a equipe não soube dividir as tarefas igualmente entre os membros, sobrecarregando alguns.

Como definido na Fase de Planejamento, os participantes deste estudo de caso foram convidados a fornecer uma nota geral para cada critério de avaliação a respeito do uso do REACT-M. Esta nota geral é definida por meio de um consenso entre a equipe. Estas notas podem ser visualizadas na Figura 8. Pode-se inferir que uma das causas para os critérios de avaliação que receberam nota 5 foi a maturidade e a mentalidade da Equipe em relação aos métodos ágeis, uma vez que em vários momentos a Equipe julgou difícil ou não necessário participar de todas as cerimônias, necessitando de uma intervenção e disseminação dos valores e princípios ágeis para que todos estivessem comprometidos e engajados ao longo do projeto.

Por fim, a partir destes resultados empíricos, um conjunto de melhorias foi identificado para ser aplicado em uma nova versão do REACT-M com a finalidade de garantir o melhor uso deste método ágil para a gerência de requisitos de software. Estes ajustes foram realizados em alguns ativos específicos que compõem o REACT-M como artefatos ou cerimônias, por exemplo. Desta forma, proporcionando um método mais flexível, eficiente, ágil e realista ao contexto atual de desenvolvimento de software. Estes ajustes podem ser conferidos no Quadro 6.

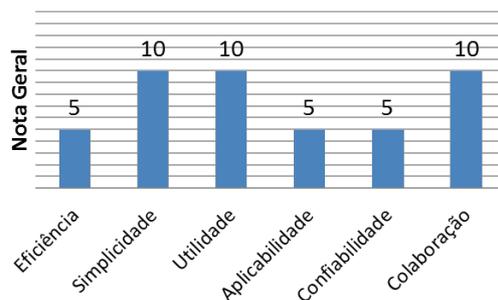


Fig. 8. Notas gerais do REACT-M por critério de avaliação.

Quadro 6. Melhorias do REACT-M.

Ativos do REACT-M	Melhorias
Critérios INVEST	<ol style="list-style-type: none"> 1. Tornar opcional, caso todos os membros da equipe participem do levantamento do <i>Backlog</i>, a validação do mesmo, tornando-se implícita. 2. Possibilitar que a equipe escolha quais critérios do INVEST irá utilizar para testar o incremento.
<i>User Story Mapping</i>	<ol style="list-style-type: none"> 1. No caso de uso de uma ferramenta que gere rastreabilidade, não exigir que seja feita manualmente. 2. O critérios de aceite gerados para um item do <i>Backlog</i> podem ficar descritos juntos, sem a necessidade de criar outro artefato para isso.
<i>Kanban</i>	<ol style="list-style-type: none"> 1. Utilizar uma ferramenta que facilite o acesso da equipe. 2. Incluir prática de atribuição de tarefas para os membros da equipe.
Formulário de Mudança	<ol style="list-style-type: none"> 1. Tornar opcional o uso deste formulário, onde a equipe deve definir quando e para qual projeto deve utilizar.
<i>Inception</i>	<ol style="list-style-type: none"> 1. Realizar a cerimônia a cada nova iteração.

5 Ameaças à Validade

O fato de avaliar o REACT-M apenas em um estudo de caso pode ser considerado como uma limitação deste estudo, uma vez que o REACT-M caracteriza-se por ser escalável e aplicável em diferentes contextos de projetos de desenvolvimento de software. Assim, outros mecanismos de triangulação seriam necessários para maximizar a validade da proposta do REACT-M, tais como *Surveys* com especialistas em métodos ágeis ou experimentos com equipes de desenvolvimento de software.

6 Conclusões e Trabalhos Futuros

O REACT-M difere-se de outras abordagens ágeis voltadas à Engenharia de Requisitos devido a sua completude e simplicidade, uma vez que a abordagem propõe um método específico para a Gerência de Requisitos que indica abordagens ágeis que podem ser usadas para executar as atividades de: I) identificar os fornecedores de requisitos; II) elaborar o *Backlog* do produto; III) avaliar o *Backlog* com a equipe técnica; IV) gerar rastreabilidade bidirecional entre os artefatos; V) revisar o *Backlog* com a finalidade de identificar possíveis inconsistências; e VI) tratar mudanças no *Backlog*. Assim como, destaca-se pela flexibilidade que fornece à equipe em algumas cerimônias. Além disso, o método preocupa-se em descrever as atividades dentro de cerimônias que possuem *time-box* definidos pelos participantes. É importante frisar que o método foi concebido para ser escalável, ou seja, foi concebido de modo que possa ser utilizado tanto em projetos pequenos como em projetos complexos de grande porte, visto que requisitos são o pilar para a construção do produto, o que torna importante a gerência destes para a boa condução do projeto. O REACT-M propõe-se a ser aplicado em diferentes contextos de projetos de desenvolvimento de software a fim de tornar a gerência de requisitos algo mais prático e menos burocrático, o que permite a equipe focar na entrega de valor ao fim do ciclo de desenvolvimento.

Por meio da condução deste estudo de caso, percebeu-se que a proposta do REACT-M atingiu ao seu objetivo, ou seja, proporcionou uma gerência dos requisitos do produto de software de forma gradual, organizada, eficiente, completa, colaborativa e ágil. Além disso, mostrou-se ser um método ágil que pode ser utilizado com outros métodos ágeis como o Scrum, bem como, um método ágil fortemente orientado a metas e centrado no cliente e nos usuários finais. Apesar do estudo de caso ter sido aplicado no início do projeto apenas em duas iterações de desenvolvimento, a equipe continuou utilizando o REACT-M após o fim do estudo de caso e até o fim do projeto, devido julgarem que o REACT-M ajudou significativamente a descoberta e o refinamento contínuo dos requisitos do produto ao longo de todo o projeto.

Como trabalhos futuros, pretende-se: a) evoluir o REACT-M para endereçar tendências da área da *Agile-RE*, por meio da condução de *Surveys* com especialistas em métodos ágeis; b) aplicar o REACT-M na indústria de software em outros contextos de desenvolvimento, tais como projetos de grande porte, distribuídos, *multistakeholder* ou em contextos de implementação dos modelos de qualidade MR-MPS-SW ou CMMI-DEV.

Neste sentido, esta pesquisa visou contribuir para a área da ER, dos métodos ágeis e da melhoria do processo de software, fornecendo um método ágil para o processo de gerência de requisitos, o qual pode ser aplicado na indústria de software por profissionais ou empresas que almejem controlar os requisitos de seus produtos a partir de métodos ágeis, bem como, servir de referência no meio científico para a realização de outros estudos com interesse sobre a *Agile-RE*.

Agradecimentos

Os autores deste trabalho gostariam de agradecer a todos os participantes do projeto SPIDER/UFPA (<http://www.spider.ufpa.br>) que participaram do estudo de caso relatado neste artigo, contribuindo ativamente para o uso e a melhoria da abordagem REACT-M. Os resultados desta pesquisa pertencem ao projeto SPIDER/UFPA.

Referências

1. ABNT-Associação Brasileira de Normas Técnicas: NBR ISO/IEC 9126-1:2003 - Engenharia de software - Qualidade de Produto Parte 1: Modelo de Qualidade. Rio de Janeiro, Brasil (2003)
2. CMMI Institute: Capability Maturity Model Integration (CMMI) for Development, Version 2.0. CMMI Institute, USA (2019)
3. Dyba, T., Dingsoyr, T.: Empirical Studies of Agile Software Development: A Systematic Review. *Information and Software Technology*, Butterworth-Heinemann Newton, MA, USA, v. 50, n. 9, pp. 833-859. (2008)
4. Inayat, I., Marczak, S., Salim, S.S., Daneva, M., Shahabuddin, S.: A systematic literature review on agile requirements engineering practices and challenges. *Journal Computer in Human Behavior*, Vol. 51, Part B, October 2014, ISSN:0747-5632, 915-929. (2014)
5. Medeiros, J.D.R.V.: An approach based on design practices to specify requirements in Agile Software Development. Tese de Doutorado. Universidade Federal de Pernambuco. Recife, PE, Brasil (2017)
6. Oliveira, S. R. B., Vasconcelos, A. M. L.: *Qualidade, Gestão e Processos de Software*. 1th Ed., Editora UFPE, ISBN 978-85-415-0733-2. (2016)
7. Parsons, D., Ryu, H., Lal, R.: The impact of methods and techniques on outcomes from agile software development projects. In: McMASTER, Tom *et al* (org.). *Organizational Dynamics of Technology-Based Innovation: Diversifying the Research Agenda*, Springer, Boston, pp. 235-249. (2007)
8. Pressman, R. S., Maxin, B. R.: *Engenharia de Software: uma abordagem profissional*. 8th Ed., McGraw-Hill. ISBN: 9788580555332. (2016)
9. Qusef, A., De Lucia, A.: Requirements Engineering in Agile Software Development. *Journal of Emerging Technologies in Web Intelligence*, VOL. 2, NO. 3. (2010).
10. Santos, K. B. C., Oliveira, S. R. B.: Um Estudo de Caso de Aplicação de um Método Ágil para Desenvolvimento de Requisitos de Software: O REACT. 21st Workshop on Requirements Engineering (WER 2018), Rio de Janeiro – Brasil. (2018).
11. Schwaber, K.: *The Enterprise and Scrum*. Redmond: Microsoft Press. (2007)
12. Silva, E.L., Menezes, E.M.: *Metodologia da Pesquisa e Elaboração da Dissertação*. 4. ed. rev. Atual – Laboratório de Ensino a Distância da UFSC (2005)
13. Sehrish, A., Shah S.A.A.: Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review. Bahria University Islamabad, Pakistan. (2017)
14. Silva, B.W.F.V., Oliveira, S.R.B.: Challenges of Requirements Management in Agile Methodologies: A Systematic Review. 16th International Conference on Information Systems & Technology Management (CONTECSI 2019). (2019)
15. Silva, B.W.F.V., Oliveira, S.R.B., Lima, A.F.A., Pinheiro, A.L.C.: REACT-M: Uma Abordagem Ágil para o Gerenciamento de Requisitos de Software. 11th Computer on the Beach (COTB 2020). (2020)
16. Sommerville, I.: *Software Engineering*. 10th Ed., Pearson Inc, Addison-Wesley. ISBN-10: 0-13-703515-2. (2019).
17. STANDISH GROUP: *Extreme Chaos Report*. The Standish Group International Inc. (2014). Disponível em: <<http://bit.ly/t1UIG>>. Acesso em: Abril/2017.
18. SOFTEX-Associação para Promoção da Excelência do Software Brasileiro: *Guia Geral: MR-MPS-SW:2020* (2020)
19. VERSIONE: The 12th annual The State of Agile Development, (2017). Disponível em: <<http://stateofagile.versionone.com>>. Acesso em: Fevereiro/2018.
20. Wainer, J.: *Métodos de pesquisa quantitativa e qualitativa para a Ciência da computação*. JAI 2007-Jornada de Atualização em Informática - Anais do XXVII Congresso da Sociedade Brasileira de Computação (2007)
21. Wiegers, K., Beaty, J.: *Software Requirements: best practices*. 3th Ed., Redmond: Microsoft Press. ISBN: 978-0-7356-7966-5 (2013)