

# Um Estudo de Caso de Aplicação de um Método Ágil para Desenvolvimento de Requisitos de Software: O REACT

Kleoson B. C. Santos<sup>1</sup>, Sandro R. B. Oliveira<sup>1</sup>

<sup>1</sup> Programa de Pós-Graduação em Ciência da Computação, Faculdade de Computação - Instituto de Ciências Exatas e Naturais - Universidade Federal do Pará, Brasil  
{brunocorrea, srbo}@ufpa.br

**Abstract.** This study presents the report about a case study on the application of REACT (Requirements Evolution in Agile Context), an agile method to support software requirements development, in a real software development environment. This case study assessed the strengths, weaknesses, opportunities, and threats related to the assets that make up REACT, namely: work products, life cycle, roles, and ceremonies. About the results, it was observed that REACT 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 development, 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 (*Requirements Evolution in Agile Context*), um método ágil para o desenvolvimento 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, a saber: artefatos, ciclo de vida, papéis e cerimônias. Dentre os resultados, observou-se que o REACT foi efetivo, simples de usar, fortemente colaborativo, *suitable*, centrado no usuário e orientado a metas. Assim, atendendo ao seu principal objetivo que é evoluir os 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 o desenvolvimento 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

O relacionamento entre a Engenharia de Requisitos (ER) e os métodos ágeis tem atraído bastante atenção dos pesquisadores nos últimos anos, como os estudos [4], [6], [10] e [11], assim como textos mais orientados para praticantes como [13], em face dos métodos ágeis terem surgido como uma ótima opção para minimizar alguns problemas enfrentados pelas abordagens tradicionais, tais como: pouco envolvimento

dos *customers*, alta mudança nos requisitos, requisitos mal especificados, dentre outros [22].

Existem inúmeros métodos ágeis utilizados para as mais diversas áreas, como Gestão de Projetos, Ciclo de Vida e Testes, tais como Scrum, XP (*eXtreme Programming*) e TDD (*Test Driven Development*). Outrossim, dada a flexibilidade que os métodos ágeis permitem para usá-los, existem diversos métodos que reúnem as melhores práticas e conceitos de outros métodos ágeis, por exemplo: ScrumS, uma adaptação do Scrum para área de *Safe Agile Development* [8]; SobA (*Story based Agile software development*), uma metodologia de desenvolvimento centrada em *user stories* [5]; Brew Model, uma técnica ágil para a priorização de requisitos [9]; assim como, métodos ágeis escaláveis como o SAFe (*Scaled Agile Framework*) [14] e o LeSS (*Large-Scale Scrum*) [12]. Isto acontece, geralmente, a fim de minimizar algum *gap* científico ou desafio deixado pelos próprios métodos ágeis atuais [15].

Neste cenário, Santos *et al.* [17] propuseram um método ágil específico para a área da ER denominado REACT (*Requirements Evolution in Agile Context*), um método que foi concebido a partir de outros métodos ágeis. É importante frisar que a proposta do REACT é de apenas desenvolver os requisitos de software, ou seja, compreende todas as atividades pertencentes ao processo de Desenvolvimento de Requisitos (DRE), uma subárea da ER responsável pela evolução, criação e interpretação 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* [20]. Em suma, este processo lida com a elicitação, análise, especificação e validação dos requisitos de um produto de software [22].

Ademais, é importante destacar que o presente estudo também corrobora com a *Agile-RE (Agile Requirements Engineering)*, a área responsável pelo relacionamento dos métodos ágeis com a ER. Esta área sugere um conjunto de boas práticas para seguir quando se trabalha com requisitos e métodos ágeis, por exemplo: os requisitos precisam ser refinados de forma iterativa e gradual ao longo do projeto [4], [10], [11], [3].

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

O restante deste estudo está estruturado da seguinte forma: Seção 2, discute sobre alguns trabalhos relacionados; Seção 3, apresenta sucintamente o REACT; Seção 4, relata o Estudo de Caso; Seção 5, apresenta e discute as limitações do estudo; e 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 conduzido por Medeiros [15], o qual propôs uma abordagem ágil para SRS (*Software Requirements Specification*). Esta abordagem foi avaliada por meio de dois estudos de caso em duas empresas de desenvolvimento de software.

Outrossim, o estudo conduzido por Boness e Harrison [2] teve como objetivo avaliar por meio de três projetos reais de desenvolvimento de software uma proposta de técnica ágil para elicitação de requisitos denominada *Goal Sketching*.

Por fim, o estudo conduzido por Daneva e Herrmann [7] apresentou resultados interessantes de um estudo de caso que contou com a participação de diversos profissionais de companhias de desenvolvimento de software da Holanda, Itália, Bulgária e Turquia, a fim de avaliar uma proposta de *Conceptual Model* para priorização ágil de requisitos. Contudo, as abordagens ágeis destes estudos não contemplaram e avaliaram o uso de todas as atividades de DRE: análise, elicitação, especificação e validação de requisitos.

### **3 REACT: Um Método Ágil para Desenvolvimento de Requisitos de Software**

O REACT (*Requirements Evolution in Agile Context*) é um método ágil com o objetivo de desenvolver 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*; e c) Análise das boas práticas constantes nos modelos de qualidade CMMI-DEV [19] e MR-MPS-SW [20], especificamente, em seus respectivos processos RD (*Requirements Development*) e DRE.

Para saber mais sobre o REACT, dois artigos foram publicados em dois eventos internacionais: o primeiro artigo de Santos e Oliveira [16] disserta sobre o MSL conduzido; enquanto que o artigo de Santos *et al.* [17] disserta sobre a criação, o funcionamento e as regras do REACT. Também é possível consultar seu Guia Prático para Uso, o qual o explica em detalhes (Disponível em: <https://goo.gl/vZPMLN>).

### **4 Estudo de Caso de Aplicação do REACT**

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.

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 [21]. 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 [21] e de Silva e Menezes [18], 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, uma vez que o REACT caracteriza-se como um método ágil para a evolução e a descoberta contínua dos requisitos de um produto. Logo, uma Equipe composta por 2 Desenvolvedores, 1 Especialista de Domínio, 1 Facilitador do REACT e 1 Cliente foi escolhida. Esta Equipe possuía a responsabilidade de conduzir um projeto de desenvolvimento de um software destinado ao ensino e aprendizado na área da

Engenharia de Software, para ser utilizado por alunos de cursos de graduação e pós-graduação na área da computação pertencentes à Universidade Federal do Pará, bem como, este produto ainda estava cercado de incertezas e o próprio cliente ainda não possuía uma ideia bem clara sobre o mesmo. Este cliente era um pesquisador da UFPA, aluno de doutorado do PPGCC/UFPA.

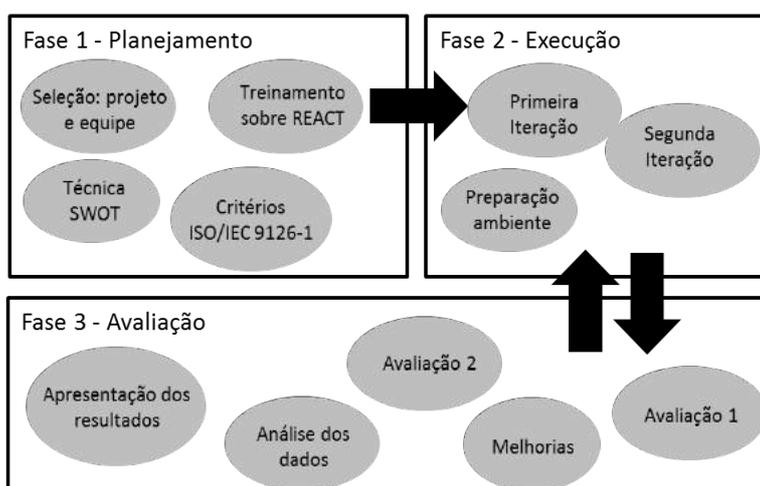


Fig. 1. Desenho do Estudo de Caso.

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.

Este estudo de caso compreendeu duas iterações de desenvolvimento com duração de 14 dias úteis cada para a aplicação do REACT. 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, 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. 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, conforme cada critério de avaliação. A escala da nota caracteriza-se da seguinte forma: a) Nota 0: para este critério o REACT não atendeu aos seus objetivos; b) Nota 5: para este critério o REACT atendeu parcialmente aos

seus objetivos; e c) Nota 10: para este critério o REACT 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. O Guia Prático mencionado na Seção 3 deste estudo foi entregue a cada participante para um melhor entendimento sobre o funcionamento do REACT.

## 4.2 Fase de Execução

É importante destacar que toda a Equipe participou de todas as cerimônias do REACT e das atividades do estudo de caso.

O primeiro passo nesta fase foi de preparar o ambiente de trabalho no qual a Equipe aplicou o REACT. A Equipe optou por utilizar recursos simples como *post-its*, pilotos, paredes e folhas A4 para criar e manter os artefatos ao longo das iterações. Assim como, os horários e os dias para realizar as cerimônias do REACT foram definidos e o espaço de trabalho reservado.

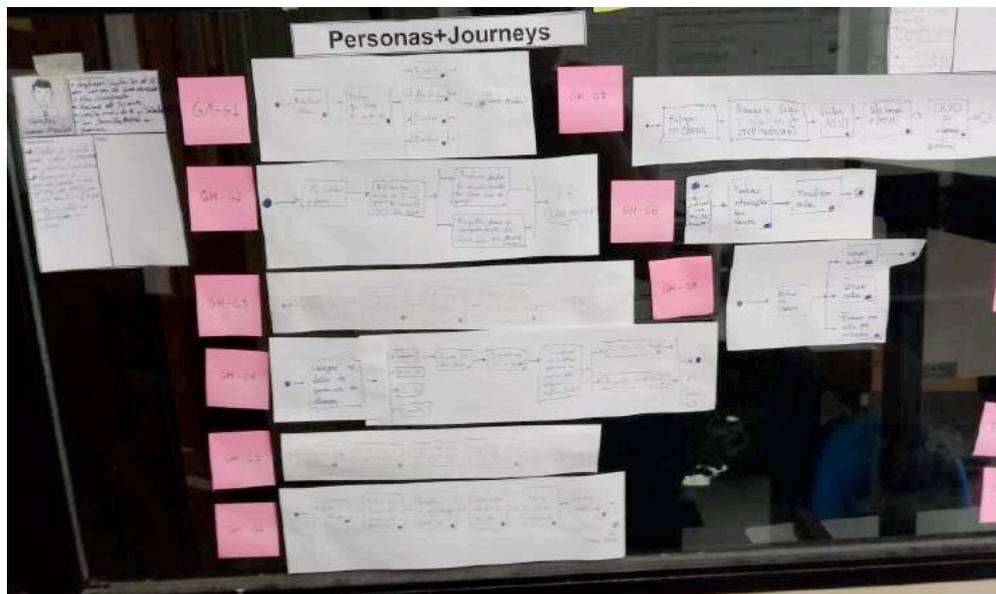
**Primeira Iteração de Desenvolvimento:** No início da primeira iteração a Equipe não possuía noção alguma de qual produto seria desenvolvido, inclusive, o próprio cliente ainda possuía muitas incertezas sobre o produto. O Facilitador conduziu a Equipe em todas as cerimônias do REACT para que seguissem as regras deste método ágil da melhor forma possível. Durante cada cerimônia, o Facilitador realizou diversas anotações de campo conforme a equipe aplicava o REACT.

Inicialmente, conforme sugere o REACT, a primeira cerimônia realizada foi a *Inception*, pela qual o problema de negócio, solução proposta, restrições e limitações do produto foram materializados em um *Canvas* de visão do produto, conforme apresentado na Figura 2(a). Posteriormente, as metas foram identificadas por meio da *Goal Sketch*, conforme Figura 2(b). Finalizando esta cerimônia com a produção das *Personas* e suas *Journeys*, visualizadas na Figura 3. Ressalta-se que o Cliente teve dificuldade em levantar as metas da *Goal Sketch*, contudo, neste momento, o Facilitador junto com o Especialista de Domínio e a Equipe tiveram que intervir fornecendo alguns exemplos de metas de outros produtos fictícios para que o Cliente tomasse como referência e definisse as metas para o seu produto, tornando esta etapa significativamente colaborativa. Outro ponto importante foi a definição das *Personas*, uma vez que permitiu à Equipe identificar o perfil, as expectativas e as metas destas *Personas* para que o produto a ser desenvolvido esteja fortemente centrado nestes usuários-chave.

Posteriormente, a segunda cerimônia realizada foi a *Story Discovery*, a qual teve o objetivo de levantar e priorizar as *users stories* e as *system stories* do produto, conforme as metas identificadas na *Goal Sketch*. A Figura 4(a) demonstra a disposição das *stories* identificadas durante a etapa de priorização. Vale ressaltar que a Equipe percebeu que esta etapa ajudaria significativamente a compor um *Product Backlog*, um artefato pertencente ao outro método ágil, o Scrum. Finalizando esta cerimônia com a produção dos *cards* do *Overall Model* em alto nível, conforme visualizado na Figura 4(b).



**Fig. 2.** (a) *Canvas do Produto*; (b) *Goal Sketch*.



**Fig. 3.** *Personas e Journeys*.



**Fig. 4.** (a) Priorização das *Stories*; (b) *Cards* do *Overall Model*.

Após isto, a cerimônia de *Refining* foi realizada. Lembrando que esta cerimônia compreende apenas as *stories* mais prioritárias para serem detalhadas, desta forma, atendendo a uma das principais práticas do REACT e da *Agile-RE: Gradual & Iterative Detailing of Requirements* [4], [10], [11] e [3]. Neste momento, os cenários de aceitação e as regras de negócio foram definidas para as *stories* mais prioritárias, conforme pode ser visualizado na Figura 5. Ressalta-se que esta cerimônia proporcionou à Equipe entender melhor sobre os requisitos do produto na visão do negócio e, principalmente, os diversos comportamentos possíveis de cada funcionalidade do produto por meio dos cenários de aceitação, prática comum no método BDD (*Behavior Driven Development*).

A próxima cerimônia realizada foi a *Modeling*, a qual teve o objetivo de identificar as interfaces internas e externas do produto, bem como, modelar os *UI Storyboards* e detalhar o *Overall Model* com mais informações. Um ponto importante informado pela Equipe foi a visão de operação e funcionamento do produto obtida a partir da elaboração dos *UI Storyboards*, os quais caracterizam-se por narrar um caminho de uso do produto, composto de *Mockups* e *Stories* logicamente encadeadas, conforme visualizado na Figura 6. Além disso, ressalta-se o fato da necessidade da Equipe entender sobre Orientação a Objetos, uma vez que o refinamento do *Overall Model* é fruto de uma análise dos requisitos, por meio da técnica de *CRC Cards*, a fim de identificar os objetos que compõem o produto de software. Desta forma, ajudando a Equipe a obter um melhor entendimento da arquitetura funcional do produto a partir dos requisitos [19].

Por fim, na cerimônia de *Inspection*, a qual teve o objetivo de verificar e validar os requisitos produzidos ao longo da execução do REACT, ressalta-se que a Equipe junto com o Cliente e o Especialista de Domínio aplicaram os critérios INVEST por meio de *check cards* em alguns níveis de requisitos produzidos, tais como *Stories* e *Persona's goals*, conforme visualizado na Figura 7. Contudo, a Equipe encontrou dificuldades em aplicar os critérios INVEST nas *Persona's goals*, tornando esta etapa um pouco arriscada, uma vez que a verificação e a validação destes requisitos podem ter sido comprometidas, visto que na *Stories* a aplicação dos critérios INVEST foi perfeitamente factível e eficiente.



Fig. 5. Business Rules e Acceptance Scenarios.

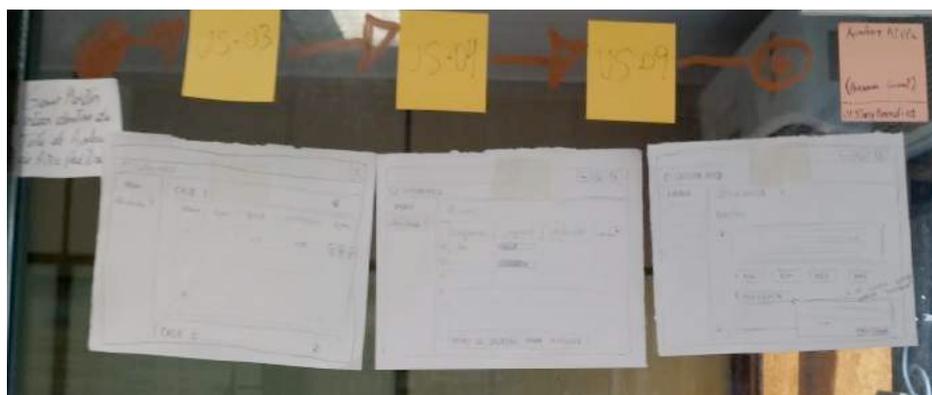


Fig. 6. UI Storyboard.



Fig. 7. Check Cards com Critérios INVEST.

Portanto, nesta primeira 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.

**Segunda Iteração de Desenvolvimento:** Antes de iniciar a segunda iteração de desenvolvimento, uma avaliação do REACT foi realizada junto à Equipe. Após isto, um conjunto de melhorias no REACT foi concebido para, então, executar a segunda iteração de desenvolvimento já contemplando estas melhorias. Estas melhorias podem ser visualizadas no Quadro 6 da subseção 4.4 deste estudo.

Basicamente, o REACT foi executado novamente desde a primeira cerimônia e seus artefatos foram evoluídos e atualizados, ou seja, novos requisitos foram descobertos e alguns requisitos existentes foram modificados para atender às reais necessidades do Cliente. Nesta segunda iteração, em suma: 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, assim, permitindo a execução das cerimônias do REACT de maneira mais eficiente e ágil.

Além disso, percebeu-se que estas melhorias proporcionaram mais flexibilidade à Equipe, uma vez que algumas etapas do REACT estavam um pouco prescritivas. Um exemplo destas melhorias ocorreu nos critérios INVEST, uma vez que a Equipe teve dificuldades de aplicar estes critérios nos vários níveis de requisitos produzidos, por exemplo: aplicar o critério SMALL nas *user stories* foi eficiente, mas aplicá-lo nas *goals* da *Persona* foi conflituoso, pois uma *goal* por natureza precisa de uma perspectiva macro. Agora a Equipe é quem define quais critérios quer utilizar na cerimônia de *Inspection*, assim, permitindo à Equipe aplicar os critérios que julgar mais relevantes para cada nível de requisito, não ficando mais “refém” apenas dos critérios INVEST para todos os níveis dos requisitos.

Outra melhoria realizada foi no ciclo de vida do REACT. Agora, a cerimônia de *Inspection* foi conduzida sempre ao fim das demais cerimônias, uma vez que percebeu-se a necessidade da validação dos requisitos de forma contínua e não negligenciada, como outrora aconteceu. Ademais, a Equipe conseguiu evoluir os requisitos do produto de forma satisfatória, ao mesmo tempo que implementava o código do produto no decorrer do projeto.

Neste sentido, a segunda iteração corroborou com um dos principais objetivos do REACT: o desenvolvimento de forma iterativa e gradual dos requisitos do produto ao longo do projeto.

### 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, 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, 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; b) a técnica de Análise SWOT foi aplicada por meio do Formulário de Avaliação SWOT (Disponível em: <https://goo.gl/vZPMLN>); 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.

#### 4.4 Sumarização dos Resultados do Estudo de Caso

Os resultados empíricos da avaliação do REACT 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.

<b>Critérios</b>	<b>S (Strengths)</b>
Eficiência	<p><b>Artefato:</b> Equipe julgou que os artefatos ajudaram muito no desenvolvimento.</p> <p><b>Artefato:</b> <i>UI Storyboard</i> ajudou a equipe a ter uma visão sobre o uso operacional e do funcionamento do software.</p> <p><b>Artefato:</b> Regras de Negócio ajudou muito a entender e detalhar as <i>stories</i>.</p> <p><b>Cerimônia:</b> Equipe julgou que a <i>Inception</i> deu uma visão geral do produto pra toda a equipe com bastante qualidade.</p> <p><b>Ciclo de vida:</b> Orientado a <i>goals</i> foi julgado um ponto extremamente forte.</p>
Simplicidade	<p><b>Papel:</b> Equipe julgou simples as responsabilidades de cada papel.</p> <p><b>Artefato:</b> <i>Canvas</i> do produto foi suficiente e eficaz para entender a visão do produto</p> <p><b>Artefato:</b> <i>Overall Model</i>, foi suficiente e fácil de criar, assim como, a maioria dos artefatos do REACT.</p>
Utilidade	<p><b>Artefato:</b> Cenários de aceitação foram bastante úteis para detalhar os requisitos.</p> <p><b>Artefato:</b> <i>Personas</i> ajudou muito na descoberta de requisitos e para guiar o desenvolvimento.</p> <p><b>Artefato:</b> <i>Personas</i> e <i>journeys</i> foram julgados os artefatos mais importantes de todos.</p>
Aplicabilidade	
Confiabilidade	<p><b>Ciclo de vida:</b> Todas as etapas e artefatos tem ligação entre si, não há <i>gaps</i>.</p>
Colaboração em Equipe	<p><b>REACT:</b> A participação de toda a equipe foi muito significativa em todas as cerimônias. Sempre discutindo e compartilhando conhecimento sobre o produto.</p>

**Quadro 3.** Fraquezas do REACT.

<b>Critérios</b>	<b>W (Weaknesses)</b>
Eficiência	<p><b>Cerimônias:</b> Custo muito alto de fazer a <i>Inspection</i> apenas uma vez.</p> <p><b>Artefato:</b> <i>Goal Sketch</i>: equipe julgou que as <i>stories</i> mais prioritárias dificultaram o desenvolvimento, devido não estarem logicamente próximas.</p>
Simplicidade	<p><b>Artefato:</b> <i>Overall Model</i>: foi julgado que deveria ser detalhado por completo na primeira iteração, o que dificultou o desenvolvimento do produto pela sua simplicidade.</p>
Utilidade	<p><b>Artefato:</b> Componentes e Interfaces: equipe não considerou muito importante para o REACT, usou-se pouco. Foi julgado inadequado para esse projeto.</p>

Critérios	W ( <i>Weaknesses</i> )
Aplicabilidade	<b>Artefato:</b> <i>Customer</i> julgou difícil de levantar as <i>business goals</i> . <b>Papel:</b> <i>Customer</i> não considerou relevante participar da criação/refinamento do <i>Overall Model</i> .
Confiabilidade	
Colaboração em Equipe	

**Quadro 4.** Oportunidades do REACT.

Critérios	O ( <i>Opportunities</i> )
Eficiência	<b>Artefato:</b> Rastreabilidade por ID (identificadores) entre os artefatos. <b>Artefato:</b> Priorização das <i>stories</i> poderia ser por <i>journey</i> . <b>Artefato:</b> Rastrear <i>stories</i> com cada <i>journey</i> . <b>Artefato:</b> Uso de uma ferramenta para ajudar a criar e manter os artefatos. <b>Ciclo de vida:</b> <i>Inspection</i> poderia ser obrigatória ao final de cada etapa do REACT. <b>Artefato:</b> <i>Goals</i> da <i>Persona</i> são mais importantes para priorizar as <i>stories</i> . <b>Artefato:</b> Mapear as jornadas com a <i>Goal Sketch</i> .
Simplicidade	
Utilidade	
Aplicabilidade	
Confiabilidade	<b>Artefato:</b> <i>Overall Model</i> : equipe considerou que precisa de mais detalhes para a criação do projeto lógico do banco de dados e para ajudar o desenvolvimento como um todo.
Colaboração em Equipe	

**Quadro 5.** Ameaças do REACT.

Critérios	T ( <i>Threats</i> )
Eficiência	
Simplicidade	
Utilidade	<b>Cerimônia:</b> <i>Inspection</i> foi julgada a menos relevante (talvez devido à participação efetiva da equipe em todas as cerimônias ou talvez devido aos critérios INVEST).
Aplicabilidade	<b>Artefato:</b> Aplicar os critérios INVEST para outros artefatos ficou muito difícil (Ex.: no <i>UI Storyboard</i> ).
Confiabilidade	
Colaboração em Equipe	<b>Papel:</b> <i>Customer</i> pode não ter muito tempo para se dedicar a todas as cerimônias do REACT com qualidade e atenção necessárias.

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. 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 com a finalidade de

garantir o melhor uso deste método ágil para o desenvolvimento de requisitos de software. Estes ajustes foram realizados em alguns ativos específicos que compõem o REACT 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.

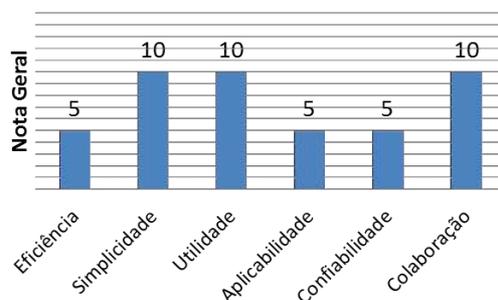


Fig. 8. Notas gerais do REACT por critério de avaliação.

Quadro 6. Melhorias do REACT.

Ativos do REACT	Melhorias
<i>Goal Sketch</i>	a) As <i>Personas's goals</i> agora possuem mais importância. Elas compõem o <i>Goal Sketch</i> . Assim, fazendo o restante do REACT ainda continuar orientado ao <i>Goal Sketch</i> , contudo, tendo as <i>goals</i> das <i>Personas</i> como referências; b) As <i>goals</i> agora são identificadas por um ID sequencial, a fim de facilitar a rastreabilidade com os outros artefatos.
Priorização das <i>Stories</i>	As <i>stories</i> agora são priorizadas pelas <i>goals</i> do <i>Goal Sketch</i> . Desta forma, garantindo uma priorização de <i>stories</i> logicamente próximas uma da outra para serem refinadas e desenvolvidas em primeiro lugar.
<i>Overall Model</i>	A equipe pode refinar com mais detalhes o <i>Overall Model</i> na cerimônia de <i>Modeling</i> , podendo ser representado, inclusive, por um diagrama de classes da UML. Deixando o modelo mais flexível.
Critérios INVEST	Podem ser substituídos. A equipe deve definir quais critérios aplicar, devendo fazer melhor sentido à realidade de cada projeto e equipe. Não é mais um padrão prescrito. Contudo, o REACT pode sugerir outros critérios de exemplos: Corretude, Ambiguidade, Conformidade e Consistência.
<i>Inspection</i> e Ciclo de Vida	Agora é realizada sempre no fim das outras cerimônias, e não mais uma única vez por iteração. Assim, garantindo uma verificação e validação dos requisitos mais eficaz, não negligenciada e constante.
Interfaces externas e internas	Uso de alguma representação em forma de modelo, ao invés de texto. Um diagrama de componentes da UML, por exemplo. Assim, facilitando uma visão geral e mais eficiente das interfaces de todo o produto e de seus componentes relacionados.

## 5 Ameaças à Validade

O fato de avaliar o REACT apenas em um estudo de caso pode ser considerado como uma limitação deste estudo, uma vez que o REACT 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, tais como *Surveys* com especialistas em métodos ágeis ou experimentos com equipes de desenvolvimento de software.

## 6 Conclusões e Trabalhos Futuros

O REACT diferencia-se de outras abordagens ágeis de apoio à ER devido a sua completude e simplicidade, uma vez que o método proporciona à equipe de desenvolvimento uma solução que atende às principais atividades da ER, ou seja, eliciação, análise, especificação e validação de requisitos. Assim como, destaca-se pela flexibilidade que fornece à equipe em algumas cerimônias. É importante dizer que o método é possivelmente escalável, ou seja, foi concebido para ser aplicado tanto em projetos pequenos como em projetos complexos de grande porte, visto que os requisitos necessitam ser descobertos e evoluídos continuamente em qualquer porte de projeto. O *time-box* das cerimônias e o tamanho da equipe é facilmente ajustável a cada cenário de desenvolvimento de software. No mais, o REACT propõe-se a ser aplicado em diferentes contextos de projetos de desenvolvimento de software, a fim de produzir especificações suficientes, eficientes e enxutas, as quais ajudem na construção de um software de maior qualidade e que atenda às reais necessidades de seus clientes e usuários finais.

Por meio da condução deste estudo de caso, percebeu-se que a proposta do REACT atingiu ao seu objetivo, ou seja, proporcionou uma evolução 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 BDD e 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 após o fim do estudo de caso e até o fim do projeto, devido julgarem que o REACT 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 para endereçar tendências da área da *Agile-RE*, por meio da condução de *Surveys* com especialistas em métodos ágeis; b) aplicar o REACT 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 DRE, o qual pode ser aplicado na indústria de software por profissionais ou empresas que almejem evoluir 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*.

## 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. Boness, K., Harrison, R.: Goal Sketching: Towards Agile Requirements Engineering. International Conference on Software Engineering Advances (ICSEA 2007). DOI: 10.1109/ICSEA.2007.36. Cap Esterel, France (2007)
3. Bjarnason, E., Wnuk, K., Regnell, B.: A Case Study on Benefits and Side-Effects of Agile Practices in Large-Scale Requirements Engineering. AREW 11 - Workshop on Agile Requirements Engineering. DOI: 10.1145/2068783.2068786. Publisher: AMC. Lancaster, United Kingdom, July (2011)

4. Cao, L.C.L., Ramesh, B.: Agile requirements engineering practices: An empirical study. *IEEE Software*, 25(1) (2008) 60–67
5. Chetankumar, P., Ramachandran, M.: SoBA: A tool support for story card based agile software development. International Conference on Software Engineering Theory and Practice, SETP-08, Orlando, Florida, USA, July 7-10 (2008)
6. Daneva, M., Inayat, I., Moraes, L., Salim, S.: A Reflection on Agile Requirements Engineering: Solutions Brought and Challenges Posed. XP 2015 Workshops, Helsinki, Finland (2015)
7. Daneva, M., Herrmann, A.: A Conceptual Model of Client-driven Agile Requirements Prioritization: Results of a Case Study. ESEM'10, September 16-17, 2010, Bolzano-Bozen, Italy (2010)
8. Esteves, R., Rodrigues, L.A., Pinto, N.A.: ScrumS: a model for safe agile development. 7th International Conference on Management of computational and collective intelligence in Digital EcoSystems. DOI: 10.1145/2857218.2857225. Publisher: ACM. Caraguatuba, Brazil, October 25 - 29 (2015)
9. Guerra, E.M., Negrão, E.C.: A Case Study for Prioritizing Features in Environments with Multiple Stakeholders. OOPSLA'11: ACM international conference companion on Object oriented programming systems languages and applications companion. DOI: 10.1145/2048147.2048187. Publisher: ACM. Portland, Oregon, USA, October 22 - 27 (2011)
10. Heikkilä, V.T., Lassenius, C., Damian, D., Paasivaara, M.: A Mapping Study on Requirements Engineering in Agile Software Development. 41 st Euromicro Conference on Software Engineering and Advanced Applications (2015)
11. 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 (2014) 915–929
12. Larman, C., Vodde, B.: Large-Scale Scrum: More with LeSS. Addison-Wesley Professional, 1 edition, ISBN-10: 0321985710. USA (2016)
13. Leffingwell, D.: Agile software requirements: lean requirements practices for teams, programs, and the enterprise. 1th Ed., Addison-Wesley Professional. ISBN-10: 0-321-63584-1 (2011)
14. Leffingwell, D., *et al.*: SAFe – Scaled Agile Framework. Disponível em: <<http://www.scaledagileframework.com/about/>>. Acesso em: Dezembro/2017 (2018)
15. 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)
16. Santos, K., Oliveira, S.: Um Estudo baseado em Evidências sobre Abordagens Ágeis de apoio ao Desenvolvimento de Requisitos de Software. XX CibSE: Ibero-American Conference on Software Engineering – Experimental Software Engineering Track. Buenos Aires, Argentina (2017)
17. Santos, K., Oliveira, S., Filho, J.: REACT: An Agile Method to Software Requirements Development. In: 15th International Conference on Information Systems and Technology Management, São Paulo-SP (2018). Disponível em: <http://contecsi.fea.usp.br/index.php/contecsi/15CONTECSI/paper/view/5760/3252>
18. 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)
19. SEI-Software Engineering Institute: Capability Maturity Model Integration (CMMI) for Development, Version 1.3. Carnegie Mellon, USA (2010)
20. SOFTEX-Associação para Promoção da Excelência do Software Brasileiro: Guia de Implementação – Parte 4: Nível D:2016 (2016)
21. 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)
22. Wiegers, K., Beaty, J.: Software Requirements: best practices. 3th Ed., Redmond: Microsoft Press. ISBN: 978-0-7356-7966-5 (2013)