

# REACT-M: uma Abordagem Ágil para o Gerenciamento de Requisitos de Software

Bleno Wilson Frankiin Vale da Silva

Programa de Pós-Graduação em Ciência da Computação  
Universidade Federal do Pará (UFPA)  
Belém, PA, Brasil  
blenofvale@gmail.com

Aline Franciele dos Anjos Lima

Programa de Pós-Graduação em Ciência da Computação  
Universidade Federal do Pará (UFPA)  
Belém, PA, Brasil  
af.lima02@gmail.com

André Luiz Coelho Pinheiro

Faculdade da Computação - Universidade Federal do  
Pará (UFPA)  
Belém, PA, Brasil  
acgrafic@gmail.com

Sandro Ronaldo Bezerra Oliveira

Programa de Pós-Graduação em Ciência da Computação  
Universidade Federal do Pará (UFPA)  
Belém, PA, Brasil  
srbo@ufpa.br

## ABSTRACT

Agile methods provide support to software engineering activities, aiming to reduce some of problems of the traditional approaches. Quality products can be developed by agile methodology in less time and budget. Requirements engineering is a building block of software development. Therefore, the search for agile practices for the evolution of requirements becomes relevant. Although several researches have been carried out on the role of requirements engineering in agile methodologies, there is still a need for studies that seek to align agile approaches to requirements development and management processes that satisfy highly accepted quality models in the market. This paper aims to establish an agile approach to support software requirements management, based on a systematic literature mapping that draws on existing approaches in agile methodologies that support requirements management activities.

## KEYWORDS

Engenharia de Requisitos, Desenvolvimento Ágil, Gerência de Requisitos, Revisão Sistemática, Modelos de Qualidade.

## 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 [1]. As abordagens ágeis popularizaram-se a partir da criação do manifesto ágil em 2001. Esse manifesto propõe novas formas de desenvolver software, fundamentados em doze princípios e quatro valores [7]. 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 [2].

Dentre as áreas de conhecimento da engenharia de software, a engenharia de requisitos possui extrema importância no ciclo de vida de projetos de software [3]. Com base nisso, modelos de qualidade para processos de software, como o MR-MPS-SW [4] e o CMMI-DEV [5], definem resultados esperados ou práticas para a implementação de desenvolvimento e gerência de requisitos.

Neste cenário, 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 em processos tradicionais, como o RUP [6]. De maneira geral, as atividades referentes a requisitos estão divididas em dois grupos: Desenvolvimento de Requisitos, que abrange as atividades de elicitação, análise, especificação e validação; e Gerência de Requisitos, que abrange as atividades de identificação, evidência de entendimento com os fornecedores de requisitos e equipe técnica, rastreabilidade, gerência de mudanças e revisão [4].

Na Engenharia de Requisitos Ágil (Agile-RE) as atividades de requisitos acontecem de forma simultânea com o desenvolvimento do software, diferente do praticado nos métodos tradicionais [8]. Por exemplo, não é preciso elicitar todos os requisitos do produto nos estágios iniciais do projeto, apenas o necessário para o entendimento inicial, e, conforme a evolução do projeto, novos requisitos são levantados, tornando o processo mais receptivo a mudanças [9].

O presente trabalho pretende, a partir de um Mapeamento Sistemático da Literatura (MSL) que identificou abordagens ágeis para a gerência de requisitos, propor um método para apoio à execução do processo de gerência de requisitos ágil. O foco em gerência de requisitos justifica-se por este trabalho ser uma extensão do trabalho produzido por Santos e Oliveira [6], que

também conduziram uma MSL para a identificação de práticas voltadas às atividades de desenvolvimento de requisitos.

### 1.1 Objetivo

O principal objetivo deste estudo é desenvolver uma abordagem ágil para apoio às atividades de gerência de requisitos e que esteja aderente ao método REACT proposto por Santos e Oliveira [6], e fundamentado em evidências empíricas presentes na literatura, fornecendo uma solução ágil para profissionais da indústria de software que almejam adotar uma abordagem ágil para a evolução dos requisitos.

### 1.2 Metodologia

Conforme as recomendações de Marconi e Lakatos [10], sobre tipos e métodos de pesquisa, a metodologia deste estudo pode ser elucidada a partir de três etapas: (1) uma etapa de levantamento de evidências empíricas disponíveis na literatura sobre abordagens ágeis de apoio ao gerência de requisitos; (2) uma etapa referente à definição de uma proposta de método ágil de apoio à gerência de requisitos; e (3) uma etapa para a avaliação da proposta em um ambiente de laboratório acadêmico de desenvolvimento de software e pesquisas científicas.

Nesse sentido, a primeira etapa deste estudo pode ser classificada quanto a seus procedimentos como uma pesquisa do tipo revisão bibliográfica, a qual será realizada por meio de uma MSL como método de pesquisa. Desta forma, ela objetiva a identificação de evidências empíricas presentes na literatura sobre abordagens ágeis de apoio às atividades de gerência de requisitos.

Já a segunda etapa deste estudo caracteriza-se por ser um estudo qualitativo, pois, em posse dos dados obtidos do mapeamento sistemático, os mesmos serão analisados por meio do método de Teoria Fundamentada a fim de embasar a proposta do método ágil de apoio às atividades de gerência de requisitos. Antes de ser validada, esta abordagem passou por uma avaliação de especialista, conduzida a partir do uso do método de Revisão por Pares, com a finalidade de obter feedback sobre as características técnicas de definição, especificação e composição do método ágil.

Por fim, na última etapa, segundo Wainer [11], este estudo caracterizar-se-á como um estudo qualitativo observacional do tipo estudo de caso, devido à interação do pesquisador junto aos sujeitos da pesquisa. Ou seja, o método ágil a ser proposto será aplicado em um ambiente de laboratório acadêmico de desenvolvimento de software e pesquisas científicas, junto a uma equipe de desenvolvimento de software que utilizará o método ágil proposto. Sendo que para avaliar a eficácia e a eficiência deste método a técnica de Entrevistas Semiestruturadas será utilizada. Esta etapa não será tratada neste trabalho.

## 2 MÉTODO REACT

O método REACT é uma proposta de método ágil para o desenvolvimento de requisitos de software, empiricamente

fundamentado [6]: em outras abordagens ágeis encontradas na literatura especializada; nas práticas da Agile-RE; e nas boas práticas constantes nos modelos CMMI-DEV e MR-MPS-SW. Esta abordagem ágil tem como objetivo evoluir as expectativas e as necessidades de negócio dos clientes ao longo do desenvolvimento de um produto de software. A ideia é que a equipe de desenvolvimento e os clientes reajam de forma mais efetiva e colaborativa frente aos desafios que permeiam a engenharia de requisitos em um projeto de desenvolvimento de software. Desta forma, produzindo especificações de software suficientes e enxutas de maneira mais eficaz, colaborativa e ágil.

É importante lembrar que o objetivo do REACT é de elicitar, analisar, especificar e validar os requisitos de software, não tendo como objetivo gerenciar requisitos, entregar software ou prover um modelo de ciclo de vida de todo o projeto de desenvolvimento de software.

## 3 MAPEAMENTO SISTEMÁTICO DA LITERATURA

O conceito de MSL, conforme o proposto por Kitchenham e Charters [12], é uma forma de estudo secundário com o objetivo de identificar e avaliar trabalhos disponíveis para uma determinada questão de pesquisa e em um certo período.

Para tal, o mapeamento sistemático faz uso de estudos publicados e validados na comunidade científica, e que são relacionados ao assunto de interesse: os estudos primários, os estudos de natureza experimental que envolvem hipóteses e os resultados obtidos com pesquisas, a partir de diferentes métodos, como surveys, estudo de caso e experimentos. Na engenharia de software os estudos experimentais são uma importante maneira de se obter maiores informações sobre novas tecnologias, metodologias e boas práticas referentes ao desenvolvimento de software [13]. O mapeamento sistemático consiste em uma pesquisa organizada da literatura, que possui como características: a abrangência, já que engloba todos ou, pelo menos, a grande maioria dos estudos relevantes a uma questão de pesquisa; não ser tendenciosa, devido a utilização de um protocolo para a revisão, que busca minimizar interesses pessoais de seus pesquisadores; e auditável, por possuir um protocolo de revisão definido a priori [13].

O principal objetivo da MSL deste estudo foi identificar trabalhos que apresentam, de maneira primária ou secundária, abordagens ágeis que possam ser agregadas a um método para a gerência de requisitos ágil. Para elucidar o objetivo, foi estabelecida a questão de pesquisa principal (QP):

- QP - Quais são as abordagens existentes nas metodologias ágeis que apoiam as atividades de Gerência de Requisitos no contexto de projetos de desenvolvimento de software?

Além da QP elaborada, um conjunto de questões de pesquisa secundárias (QS) foram estabelecidas com o objetivo de esclarecer detalhes importantes que o mapeamento sistemático procura identificar e estão relacionadas diretamente com a questão de pesquisa principal [15]. As QS estabelecidas são:

- QS1 - Quais as atividades de Gerência de Requisitos são apoiadas pelas abordagens encontradas?
- QS2 - Caso a abordagem seja uma prática ou técnica, ela pertence a algum método ágil existente? Qual?
- QS3 - Qual o contexto de aplicação da abordagem encontrada?
- QS4 - Quais os ativos (papéis e artefatos) envolvidos nessas abordagens?

Após a coleta dos estudos nas fontes de pesquisa, os trabalhos foram selecionados conforme os procedimentos: I) Definir os critérios de inclusão e exclusão, a fim de nortear os pesquisadores na avaliação dos trabalhos, além de auxiliar no rigor da pesquisa e reduzir avaliações tendenciosas [12]; II) Selecionar estudos primários a partir da leitura de título, resumo e palavras-chaves, com o intuito de eliminar estudos que claramente não estão alinhados ao objetivo desta mapeamento sistemático; III) Realizar a leitura integral dos estudos primários selecionados. Os resultados da MSL desenvolvida neste estudo encontram-se em [24].

A MSL foi executada entre janeiro de 2016 a dezembro de 2017, justificado, por este ser o período de tempo não contemplado pelo trabalho de Santos e Oliveira [6], posteriormente os resultados de ambos os trabalhos foram concatenados, devido os objetivos das MSL serem equivalentes: buscar na comunidade científica abordagens ágeis para a execução das atividades da engenharia de requisitos. Com o resultado, um total de 177 estudos primários foram selecionados, dos quais foram possíveis extrair informações para responder as questões de pesquisa de interesse deste estudo.

Foram identificadas 35 abordagens ágeis para suporte às atividades de gerência de requisitos em um ambiente ágil, conforme ilustrado na Tabela 1. Ressalta-se que algumas das abordagens encontradas não são específicas para requisitos, mas podem ser adaptadas a um processo ágil de gerência de requisitos.

Tabela 1: Abordagens coletada para Gerência de Requisitos.

QP - Quais são as abordagens existentes nas metodologias ágeis que apoiam as atividades de Gerência de Requisitos no contexto de projetos de desenvolvimento de software?			
Cód	Nome da Abordagem Ágil	Estudos Primários (ID)	Categoria
A01	<i>CBRank (Case-Based Ranking)</i>	[EP19]	Técnica
A02	<i>Discover to Deliver</i>	[EP05]	
A03	<i>FuzzyHCV</i>	[EP19]	
A04	<i>HYPEX (Hypothesis Experiment Data-Driven Development)</i>	[EP05]	
A05	<i>Human Stories</i>	[EP14]	
A06	<i>Goal Sketching</i>	[EP19]	
A07	<i>Invest (Independent, Negotiable, Valuable, Estimable and Small)</i>	[EP14]	
A08	<i>Lean canvas</i>	[EP17]	

QP - Quais são as abordagens existentes nas metodologias ágeis que apoiam as atividades de Gerência de Requisitos no contexto de projetos de desenvolvimento de software?			
Cód	Nome da Abordagem Ágil	Estudos Primários (ID)	Categoria
A09	<i>MMFs (Minimal Marketable Features)</i>	[EP19]	Método
A10	<i>QIP(Quality Improvement Paradigm)</i>	[EP13]	
A11	<i>Personas</i>	[EP22], [EP07]	
A12	<i>Planing Poker</i>	[EP13]	
A13	<i>User Story Mapping</i>	[EP05]	
A14	<i>Value Proposition Design</i>	[EP05]	
A15	<i>AAOM (Agile Agentoriented Modelling)</i>	[EP08], [EP18]	
A16	<i>ASD (Adaptive Software Developmen)</i>	[EP11], [EP21]	
A17	<i>DSDM (Dynamic Systems Development Method)</i>	[EP03], [EP11], [EP21]	
A18	<i>eXtremeProgramming</i>	[EP03], [EP06], [EP11]	
A19	<i>FDD (Feature Driven Development)</i>	[EP11]	
A20	<i>Hibrid Waterfall</i>	[EP10]	
A21	<i>Kanban</i>	[EP06], [EP11]	
A22	<i>Lean</i>	[EP06]	
A23	<i>QUS (Quality User Story)</i>	[EP02], [EP14], [EP15], [EP16]	Práticas
A24	<i>SAFe (Scaled Agile Framework)</i>	[EP06], [EP07], [EP10]	
A25	<i>Scrum</i>	[EP03], [EP06], [EP07], [EP09], [EP11], [EP13], [EP21], [EP22]	
A26	<i>Water-Scrum-Fall</i>	[EP07]	
A27	<i>Automatic Generation of Traceability Links</i>	[EP12]	
A28	<i>Overall Model</i>	[EP11]	
A29	<i>Inception Deck</i>	[EP23]	
A30	<i>Lightweight Traceability</i>	[EP09]	
A31	<i>Plan by Feature</i>	[EP11]	
A32	<i>Product Backlog</i>	[EP03], [EP06], [EP13]	
A33	<i>Review meetings</i>	[EP11], [EP21]	
A34	<i>TCR (Test Cases as Requirements)</i>	[EP20]	
A35	<i>User Stories</i>	[EP08], [EP14], [EP15], [EP16], [EP22]	

#### 4 MÉTODO REACT-M

Após a seleção das abordagens da MSL consideradas mais relevantes para o estudo, foi contruido um fluxo de atividades

que mesclam as atividades de desenvolvimento de requisitos proposta pelo método REACT com as atividades para gerência de requisitos. É importante ressaltar que as atividades de gerência de requisitos identificadas foram baseadas nos processos de Gerência de Requisitos e Requirements Management sugeridos, respectivamente, pelos modelos de qualidade MR-MPS-SW [4] e o CMMI-DEV [5].

Foi necessário levantar um conjunto de papéis que representam os indivíduos envolvidos na execução do método, sempre garantindo o alinhamento com o proposto pelo método REACT. Desta forma, os papéis envolvidos no REACT-M são:

- **Customer:** é o cliente para o qual o produto de software está sendo desenvolvido. Seu objetivo é tomar decisões sobre os requisitos ao longo do projeto;
- **Domain Expert:** é o representante do *customer*, o qual possui uma grande *expertise* em determinadas áreas de domínio dentro do negócio do *customer*. Seu objetivo é ajudar o *customer* e o *team* a desenvolver um produto que realmente atenda às necessidades do negócio;
- **Team:** é a Equipe de desenvolvimento responsável por evoluir os requisitos do produto ao longo do projeto;
- **Facilitator:** é uma espécie de *coach* responsável por guiar todos os envolvidos a aplicar o REACT-M da melhor forma possível em cada contexto de projeto de software.

A próxima etapa da construção do método foi selecionar as abordagens que seriam utilizadas para contemplar as atividades de gerência de requisitos. As seleções das abordagens foram pautadas em características como: I) Facilidade na implementação; II) Uso de artefatos de simples acesso; e III) Abordagens que já possuam boa aceitação pelo mercado. A seguir, cada atividade de gerência de requisitos será descrita.

#### 4.1 Identificar Fornecedores de Requisitos

Essa atividade busca definir estratégias para descobrir os potenciais fornecedores dos requisitos para melhor entender o problema que o produto deve resolver, bem como os objetivos que os usuários esperam alcançar com o seu uso. É importante entender que o fornecedor não é sinônimo de customer [16]. O customer que demanda o produto pode ser considerado como fornecedor de requisitos, mas outros indivíduos que serão afetados pelo produto, direta ou indiretamente, também devem ser levados em consideração.

Para realizar esta atividade foi utilizada uma adaptação da técnica de Personas, a fim de mapear as características dos fornecedores de requisitos, bem como suas expectativas e objetivos com o produto. Diferente da técnica original, o REACT-M propõe que sejam definidas personas que representem os customers do produto, além de personas que representem usuários-chaves. Desse modo, as expectativas dos Customers serão mapeadas e terão maior visibilidade. É interessante que todos os papéis estejam envolvidos para melhor idealizar as personas. Um exemplo de persona pode ser visto na Figura 1.

#### 4.2 Elaborar Backlog de Tarefas

Através das propostas definidas no método REACT, users storeis são geradas e priorizadas. Este conjunto de users stories constitui o backlog do produto. Uma atividade importante da gerência de requisitos é manter esse backlog gerenciado a fim de controlar mudanças ou impedimentos que venham a acontecer durante o projeto.

O REACT-M propõe a utilização adaptada do quadro Kanban [18]. Nesta adaptação teremos dois Kanban para gerenciar as users stories. O primeiro de responsabilidade do Domain Expert e que possibilita uma visão do negócio, permitindo visualizar as user stories que ainda serão desenvolvidas, as que estão sendo desenvolvidas dentro do ciclo e as que já foram entregues para o customer. Um exemplo de Kanban do Domain Expert Kanban pode ser visto na Figura 2.

Marina – UX Design 	Perfil - 27 anos - Trabalha 8hrs em uma fábrica de software - Trabalha muito tempo sozinha - Gosta de música alternativa - Gosta de ouvir podcasts - marina@email.com
O que faz - Ouve música alternativa - Procura novos artistas/estilos de música - Recebe e envia para os amigos sugestões de artistas/músicas - Ouve playlists sobre notícias	O que espera 1. Personalizar playlists 2. Obter recomendação de música de acordo com as que está ouvindo 3. Compartilhar música com amigos 4. Organizar músicas por estilo

Figura 1: Exemplo de Persona.

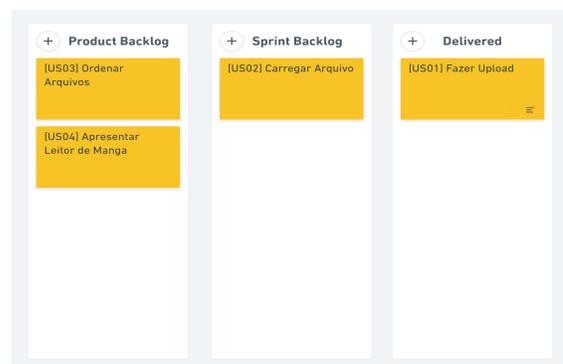


Figura 2: Exemplo Kanban do Domain Expert.

O segundo quadro Kanban proposto tem o objetivo de gerenciar as user stories que estão sendo desenvolvidas dentro do ciclo. O Team deve mover as user stories que estão no Sprint Backlog do Kanban do Domain Expert para a seção Sprint Backlog do Kanban do Team, e definir para cada uma quais são as tarefas que devem ser realizadas para garantir a sua entrega

no final do ciclo de desenvolvimento. As tarefas levantadas devem ser colocadas na seção To Do do Kanban do Team. Após o início do ciclo, o membro responsável pela execução de uma tarefa deve movê-la para a seção In Progress. Ao terminar a execução, deve-se mover a tarefa para a seção Testing. O membro responsável por testar uma tarefa deve realizar os testes necessários e, no caso de encontrar algum defeito, deve-se abrir uma nova tarefa descrevendo o problema encontrado e colocá-la na seção To Do, para que possa seguir novamente o fluxo de trabalho até a correção. No caso da tarefa não possuir defeitos ou seus defeitos terem sido corrigidos, ela deve ser movida para a seção Done do quadro Kanban indicando que está finalizada. Um exemplo do Kanban do Team pode ser visto na Figura 3.

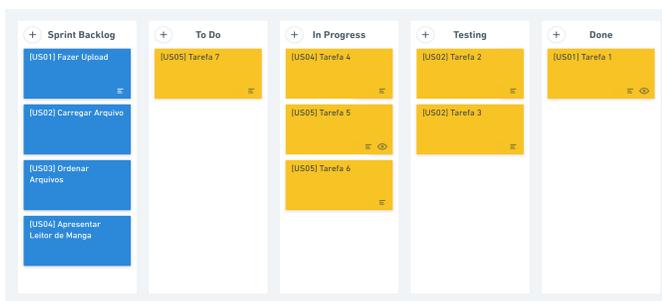


Figura 3: Exemplo de Kanban do Team.

### 4.3 Avaliar Backlog com a Equipe Técnica

O backlog do produto deve ser validado pelo Team para que este indique se os requisitos possuem informações de negócio suficientes para entrar no ciclo curto de desenvolvimento proposto pelo REACT-M. Para essa tarefa é interessante que o Team defina critérios de qualidade que os itens do backlog devem possuir.

Para essa atividade duas abordagens foram selecionadas. O conceito de Definition of Ready defendido pelo Scrum será utilizado para servir como um acordo entre o Domain Expert e o Team do que se espera que os requisitos possuam de informação para poder entrar em um ciclo de desenvolvimento[20]. Para ajudar o Team a avaliar se os requisitos estão dentro do esperado, será utilizada a técnica INVEST, que define um conjunto de critérios que precisam ser atendidos para que um requisito tenha um nível aceitável de qualidade [19]. Os critérios INVEST são apresentados a seguir:

- **INDEPENDENT**: verifica se o requisito é independente de outros para a sua implementação;
- **NEGOTIABLE**: analisa se foi definida uma prioridade do que agrega mais valor para o *customer*;
- **VALUABLE**: verifica se o requisito possui ligação com os objetivos dos *customers*;
- **ESTIMABLE**: verifica se é possível estimar o tamanho do esforço para a implementação;
- **SMALL**: verifica se o requisito é pequeno suficiente para ser implementado em uma iteração;

- **TESTABLE**: verifica se o requisito possui cenários de aceitação definidos.

Para facilitar o entendimento, o REACT-M propõe a conversão dos critérios em forma de perguntas que os membros do Team devem reponder para um requisito a fim de avaliar a sua qualidade. Um exemplo de como os critérios podem ser aplicados é apresentado na Figura 4.

DEFINITION OF READY	
<input type="checkbox"/>	A user story pode ser implementada independente de outras users stories?
<input type="checkbox"/>	A user story foi priorizada pelo cliente?
<input type="checkbox"/>	A user story reflete a expectativa do cliente?
<input type="checkbox"/>	A user story agrega valor para o negócio do cliente?
<input type="checkbox"/>	É possível estimar o esforço de implementação da user story?
<input type="checkbox"/>	A user story é pequena o suficiente que não pode ser partida em mais de uma?
<input type="checkbox"/>	Através dos critérios de aceite é possível gerar os cenários de teste?

Figura 4: Exemplo de avaliação dos requisitos com o INVEST.

O *Team* também tem a responsabilidade de estimar o esforço necessário para a implementação das *users stories*, levando em consideração questões técnicas e a experiência que os membros possuem. Além disso, é importante frisar que para estimar é necessário levar em consideração todas as tarefas do desenvolvimento. Para atender esta atividade, é sugerido o uso da técnica *Planning Poker*.

O *Planning Poker* é uma técnica do Scrum que permite ao Team gerar estimativas rapidamente. Nessa técnica não é realizada a estimativa por horas de trabalho, as estimativas são feitas utilizando user story points que representam um valor abstrato de tamanho [21]. A ideia do *Planning Poker* é pontuar as *users stories*, onde cada membro da equipe possui um baralho de 13 cartas numeradas com a sequência de Fibonacci. Neste baralho, cada valor contido na carta corresponde ao story point que o membro julga necessário para implementar a estória que está sendo estimada. A seguir são apresentados os passos da dinâmica do *Planning Poker*.

- Primeiro, escolhe-se uma *user story* que a estimativa se aproxime de 3 *story points* para servir como referência. A partir desta estória serão feitas estimativas para outras estórias seguindo o *Planning Poker*;
- O *Domain Expert* lê para o *Team* a *user story* e contextualiza a respeito do seu valor de negócio;
- Cada membro do *Team* define individualmente quantos *story points* são necessários para implementar e seleciona a carta correspondente;
- Quando todos os membros tiverem escolhido uma carta, todos viram ao mesmo tempo;
- Caso os valores sejam muito divergentes, cada membro apresenta uma justificativa geralmente do valor mais alto ao mais baixo;
- Então, é feita outra rodada até que as estimativas de esforço cheguem a uma convergência, isto é, todas as estimativas sejam aproximadamente a mesma para cada estória;
- O resultado será o valor que, ao final, tiver maior ocorrência ou a média entre estes valores finais.

### 4.4 Gerar Rastreabilidade

Em um projeto ágil a rastreabilidade é tratada como mais um elemento que visa aproximar aquilo que o usuário pede daquilo que ele recebe. Ele tem que ser trabalhado de uma maneira que dispense o uso de planilhas ou qualquer alimentação manual de dados. Para o REACT-M foi proposto realizar a rastreabilidade dos requisitos tanto de mesmo nível ou nível diferente a partir da adaptação da abordagem de User Story Mapping, que é um método muito utilizado para criar uma visão geral do produto, devido a estratégia que possui de gerar conexões entre as informações relacionadas a uma determinada ideia [22]. O REACT-M propõe-se utilizar da mesma estratégia de conexão do User Story Mapping para gerar a rastreabilidade entre os artefatos gerados no método.

A sugestão para o REACT-M é que os artefatos gerados pelo método, como: user story, critérios de aceite, protótipos, dentre outros; possuam um identificador pelo qual será possível fazer a relação entre eles. Exemplos de como pode ser feita a rastreabilidade no REACT-M pode ser visto na Figura 5.



Figura 5: Exemplo de Rastreabilidade com User Story Mapping

### 4.5 Corrigir Inconsistências

Esta tarefa visa verificar e avaliar os vários níveis de requisitos produzidos ao longo da execução do método, com o objetivo de analisar se os requisitos produzidos estão corretos, testáveis, pequenos suficientes e se estão alinhados às reais necessidades do customer em relação ao produto. Para realizar esta atividade o REACT-M propõe novamente o uso dos conceitos apresentados na atividade Avaliar Backlog com a Equipe Técnica, onde é proposto que a qualidade dos requisitos seja avaliada a partir do acordado entre o Domain Expert e Team, utilizando a técnica de Definition of Ready e os critérios INVEST.

### 4.6 Identificar Mudança e Analisar Impacto

Mudanças são uma realidade no desenvolvimento de software. Estar disposto a receber mudanças e atento para respondê-las rapidamente, certamente irá gerar mais valor para o customer. Para isso, os membros envolvidos no projeto devem utilizar-se de estratégias para receber e tratar as mudanças que irão aparecer ao longo do desenvolvimento. No entanto, existem projetos que mudanças podem causar grandes impactos, como sistemas

bancários ou para atendimento médico. Deste modo, é necessário que se faça uma estratégia que não só acate a mudança, mas também realize uma análise de impacto dentro do escopo do projeto.

O REACT-M busca um equilíbrio a fim de propor uma gestão de mudança mais estruturada, porém sem burocratizar esta atividade crucial para o bom andamento do projeto. Para isso, é proposta a utilização de conceitos oriundos dos métodos Scrum e Water-Scrum-Fall para tratar as mudanças. Os eventos do Scrum permitem que, ao longo do ciclo de desenvolvimento, o cliente possa inclusive entender aquilo que está sendo construído e solicitar mudanças [20]. Já os conceitos do Water-Scrum-Fall determinam diretrizes que devem ser feitas quando uma mudança surge [23]. O método REACT-M propõe que a mudança seja tratada dentro das seguintes etapas:

- Não incluir mudanças dentro do ciclo atual;
- Atualizar o *backlog*, inserindo as novas mudanças e realizando a análise dos impactos;
- Realizar nova avaliação do *backlog* com a Equipe Técnica;
- Ter um formulário de controle de mudança leve, se necessário.

Essas diretrizes definem que a mudança deve passar pelos fluxos de atividades propostos até o momento para entendê-las, avaliá-las e rastreá-las dentro do escopo do projeto. Deste modo, os envolvidos conseguirão entender a mudança e seus impactos de maneira adequada, bem como será mais fácil realizar o gerenciamento. O REACT-M também sugere o uso de uma tabela de histórico que permite uma visualização rápida das mudanças ocorridas durante o projeto, como pode ser visto na Figura 6.

N°	Data da Solicitação	Responsável	Descrição	User Story		Esforço
				Impacto	Novas	
RQM01	28/05/2019	Nome do Responsável	Incluir validação por token via SMS para efetivar as transferências por DOC e TED.	US02; US03	US04; US05	20
RQM02	02/06/2019	Nome do Responsável	Inclusão da lista de favoritos para a Transferência via TED.	US02	US06	8
RQM03	12/07/2019	Nome do Responsável	Incluir opção Ajuda nas funcionalidade de Poupança Programada	-	US07	3

Figura 6: Exemplo Tabela de Mudanças.

Para facilitar o uso por organizações e indivíduos que estejam interessados em adotar o método para a evolução dos requisitos em ambiente ágil, foi elaborado um guia de utilização que objetiva explicar o REACT-M como um passo-a-passo, dando exemplos de uso das abordagens para contextualizar e facilitar o entendimento de terceiros sem a necessidade da ajuda de alguém que tenha certo domínio no método.

## 5 AVALIAÇÃO DO MÉTODO PROPOSTO

A avaliação do método REACT-M proposto para a implementação da gerência de requisitos ágil ocorreu de maneira sistemática, utilizando a técnica de Revisão por Pares. O fluxo das atividades realizadas durante a revisão por pares foi: I) identificação do avaliador; II) definição dos critérios de

classificação; e III) avaliação do método por meio da técnica de revisão por pares.

A revisão por pares foi realizada com o objetivo de verificar se o método REACT-M possui o conteúdo necessário para que possa ser implementado em um ambiente real de desenvolvimento de produtos de software. Outrossim, foi avaliar se as abordagens propostas no método contemplam os resultados esperados nos processos de gerência de requisitos definidos nos modelos de qualidade CMMI-DEV e MR-MPS-SW. Assim, deve-se garantir que a adoção da solução proposta, não apenas busque utilizar conceitos ágeis, mas também cumpre com os padrões de qualidade que são respeitados pelo mercado de software.

A primeira etapa da revisão por pares foi identificar um revisor que tivesse conhecimento e experiência nos dois modelos de qualidade (CMMI-DEV e MR-MPS-SW), e que também possuísse experiência em métodos ágeis. Deste modo, considerou-se necessário que o avaliador tivesse as seguintes características: nível de conhecimento em modelos de referência de processo de software (CMMI-DEV e MR-MPS-SW); experiência na implantação ou avaliação de modelos para melhoria do processo de software em organizações; tempo de experiência em implantação de modelos para melhoria do processo de software; certificação em modelos para melhoria do processo de software; nível de conhecimento em métodos ágeis; tempo de experiência na implantação de métodos ágeis em organizações.

O avaliador escolhido atendeu todos os critérios de seleção mencionados, pois o mesmo possui um alto nível de experiência dentro da área abordada, atuando a mais de cinco (05) anos na implementação de modelos de qualidade em organizações, tais como: CMMI, MPS.BR, CERTICS, ISSO/IEC 12207, MPT.BR. Este revisor possui certificações nos modelos MPS.BR e CMMI-DEV, e também atua como avaliador de processos de software. Além disso, possui experiência de mais de cinco (05) anos fornecendo consultorias para organizações que estão migrando seus processos de métodos tradicionais para ágeis.

A segunda etapa teve foco na definição dos critérios de avaliação a serem utilizados pelo avaliador, para que possa expressar as suas críticas a respeito do REACT-M, com o intuito de ajudar a identificar as fraquezas e o que pode ser aprimorado para possibilitar uma melhor adoção do método. Os critérios definidos foram:

- **Técnico Alto (TA):** indica que foi encontrado um problema em algum item que no caso de não ser corrigido, comprometerá o entendimento da atividade;
- **Técnico Baixo (TB):** indica que foi encontrado um problema em algum item que não afetará o entendimento da atividade, porém é conveniente a correção;
- **Editorial (E):** indica que foi encontrado algum erro gramatical ou que o texto pode ser melhorado;
- **Questionamento (Q):** Indica que houve dúvida em algum ponto do conteúdo;
- **Geral (G):** indica comentários gerais feitos pelo avaliador em relação ao conteúdo revisado.

O documento entregue para a revisão foi o guia de utilização elaborado para o método REACT-M. O guia segue o mesmo fluxo de atividades apresentados na Seção 3, porém estruturado de uma maneira mais adequada para a utilização do método por interessados. Vale ressaltar que o conteúdo apresentado neste artigo já está em conformidade com as correções solicitadas pelo revisor, tornando o método apresentado em sua segunda versão.

Os problemas identificados pelo revisor foram registrados no formulário de revisão por pares. Ao fim, o revisor devolveu o Termo de Confidencialidade e o Formulário de Revisão por Pares com as devidas observações. Foi identificado um (1) problema técnico alto, dois (2) problemas técnicos baixos e um (1) problema editorial. O avaliador não classificou nenhum problema como Questionamento (Q) ou Geral (G).

O revisor identificou um problema TA, que consistia na falta de descrição prática para algumas atividades propostas no guia do REACT-M, recomendando que fossem apresentados exemplos mais fáceis de como essas atividades podem ser executadas em ambiente real de organizações de software, como por exemplo, apresentar o passo-a-passo da execução para a construção de um produto de software, para um público específico, tal como: aplicativo de Internet Banking, streamer de música, etc. A utilização de exemplos deste tipo possibilitará que o leitor consiga melhor relacionar o seu conjunto de clientes reais, e como pode adotar o método em seu processo de gerência de requisitos.

O primeiro problema classificado como TB foi referente à atividade de Identificar Fornecedores de Requisitos. Esta atividade propõe a utilização da técnica de personas, porém não estava claro no texto como seriam identificados os fornecedores de requisitos, tendo em vista que a técnica original personifica usuários-chaves que não são necessariamente os fornecedores de requisitos de um produto. A sugestão do revisor foi alterar a descrição da atividade para deixar de maneira clara ao leitor que é utilizada uma adaptação da técnica de Personas.

O guia não deixava explícito como seria feita a rastreabilidade da tarefa de defeito com a tarefa original, este ponto foi definido pelo revisor como o segundo problema TB. Foi recomendada pelo revisor a atualização das descrições das atividades de Gerar Rastreabilidade e Elaborar Backlog de Tarefas. A primeira por ser a atividade específica que trata de como a rastreabilidade será feita no REACT-M, e a segunda por descrever o quadro Kanban do Team, que é a proposta do método para gerenciar as tarefas que serão executadas dentro do ciclo de desenvolvimento, e, por isso, é onde está a descrição de como as tarefas de defeito são tratadas no método.

Por fim, o problema editorial identificado pelo revisor diz respeito a erros ortográficos dentro do guia que devem ser ajustados para adequar o guia de utilização ao formalismo esperado a um documento deste tipo.

## 6 RESULTADOS ESPERADOS

A próxima etapa deste estudo é validar o REACT-M para analisar a aplicabilidade da proposta em um ambiente real de

desenvolvimento de software, a fim de identificar as forças, as fraquezas, as oportunidades de melhoria e as ameaças relacionadas ao uso do método. Para isto, um estudo de caso foi escolhido como método de pesquisa por se tratar de uma abordagem onde o pesquisador interage com os indivíduos que estão utilizando o método por meio de entrevistas e conversas programadas, a fim de levantar as avaliações ou opiniões destes sujeitos [11]. A partir do feedback dos envolvidos nesta avaliação, pretende-se tabular as críticas e realizar correções, a fim de melhorar características do método e torná-lo mais adequado à realidade das empresas de desenvolvimento de software.

## 7 CONCLUSÃO

O trabalho desenvolvido concentrou-se na investigação de abordagens existentes para a definição de um método para apoio à gerência de requisitos em ambientes de desenvolvimento de software ágil. Esta investigação foi conduzida a partir de um mapeamento sistemático da literatura, que forneceu um arcabouço empírico de abordagens ágeis para apoio às atividades de gerência de requisitos, o qual permitiu entender melhor o relacionamento entre métodos ágeis e gerência de requisitos. Observou-se que a maioria das abordagens ágeis coletadas não são específicas para requisitos, muitas são práticas e técnicas genéricas ou pertencentes a outros métodos ágeis, as quais apenas ajudam na evolução dos requisitos. No entanto, também foram identificadas abordagens específicas para a área de gerência de requisitos, as quais podem auxiliar em atividades como: gestão de mudança, rastreabilidade, análise de impacto, dentre outras.

Com o embasamento obtido dos estudos coletados no mapeamento sistemático, foi proposto um método ágil para a gerência de requisitos, que também está fundamentada nas boas práticas constantes nos modelos CMMI-DEV e MR-MPS-SW. Além disso, o método concebido também visa estar alinhado ao método proposto para desenvolvimento de requisitos por Santos e Oliveira [6]. Assim, fornece para a comunidade uma metodologia para a implantação das atividades de desenvolvimento e gerência de requisitos em projetos de desenvolvimento de software dentro de organizações que almejem implementar a melhoria do processo de software através de métodos ágeis.

Como trabalho futuro, encontra-se em curso a aplicação do método ágil em um experimento com foco no desenvolvimento de um projeto de software no laboratório de pesquisa e desenvolvimento da equipe responsável pela pesquisa, onde será realizada a análise dos resultados obtidos com a aplicação do método e serão identificados os indicadores de eficácia e eficiência do método.

## REFERÊNCIAS

[1] Versione. The 10th annual The State of Agile Development, 2015. Disponível em: <http://stateofagile.versionone.com>. Acesso em: 20 Nov. de 2018.

[2] A. Qusef and A. de Lucia, 2010. Requirements Engineering in Agile Software Development. *Journal of Emerging Technologies in Web Intelligence*, VOL. 2, NO. 3.

[3] R.S. Pressman and B.R. Maxin, 2016. *Engenharia de Software: uma abordagem profissional*. 8th Ed., McGraw-Hill. ISBN: 9788580555332.

[4] SOFTEX - Associação para Promoção da Excelência do Software Brasileiro, 2016. *Melhoria do Processo de Software Brasileiro (MPS.BR) - Guia Geral*.

[5] SEI, 2010. *Capability Maturity Model Integration (CMMI) for Development*, Version 1.3. Carnegie Mellon, USA.

[6] K.B.C. Santos and S.R.B. Oliveira, 2017. *Uma Proposta de Método Ágil para apoio ao Processo de Desenvolvimento de Requisitos no Contexto de Projetos de Desenvolvimento de Software*. Dissertação de Mestrado, PPGCC/UFGA, Pará.

[7] K. Schwaber and M. Beedle, 2002. *Agile Software Development with Scrum*. 1th Edition, Prentice Hall.

[8] M. Daneva, I. Inayat, L. Moraes and S.A. Salim, 2015. *Reflection on Agile Requirements Engineering: Solutions Brought and Challenges Posed*. XP 2015 Workshops, Helsinki, Finland.

[9] V.T. Heikkilä, C. Lassenius, D. Damian and M.A. Paasivaara, 2015. *Mapping Study on Requirements Engineering in Agile Software Development*. 41st Euromicro Conference on Software Engineering and Advanced Applications.

[10] M.A. Marconi and E.M. Lakatos, 2003. *Fundamentos de Metodologia Científica*. 5th Ed., Editora Atlas S. A., ISBN 85-224-3397-6.

[11] J. Wainer, 2007. *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.

[12] B. Kitchenham and S. Carters, 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*, Technical Report EBSE-2007-01, Department of Computer Science Keele University, Keele.

[13] S. Mafra and G. Travassos, 2006. *Estudos Primários e Secundários apoiando a busca por Evidencia em Engenharia de Software - Relatório Técnico: RT-ES-687/06 -COPPE/UFRJ - Rio de Janeiro*.

[14] G. Santos, 2010. *Mapeamento sistemático: Minicurso. Simpósio Brasileiro de Qualidade de Software - SBQS 2010, Belém - PA*.

[15] M. Palomino, A. Dávila, K. Melendez and M. Pessoa, 2016. *Agile Practices Adoption in CMMI Organizations: A Systematic Literature Review*. In: Mejla J., Munoz M., Rocha Á., San Feliu T., Pena A. *Trends and Applications in Software Engineering*. CIMPS. *Advances in Intelligent Systems and Computing*, vol 537. Springer, Cham.

[16] L. Zamudio, L., A. Aguilar, C. Tripp and S. Misra, 2017. *A Requirements Engineering Techniques Review in Agile Software Development Methods*. International Conference on Computational Science and Its Applications -ICCSA - Trieste, Italy.

[17] M. Ware, 2008. *Peer review: benefits, perceptions and alternatives*. Publishing Research Consortium, London, Britain.

[18] M. Alqudah and R. Razali, 2017. *A comparison of scrum and Kanban for identifying their selection factors*. International Conference on Electrical Engineering and Informatics (ICEEI). Langkawi, Malaysia, 2017.

[19] E. Bjarnason and M. Borg, 2017. *Aligning Requirements and Testing: Working Together toward the Same Goal* IEEE Software (Volume: 34, Issue: 1).

[20] D. Silva, E. Oliveira, E. Canedo and H. Martins, 2016. *Application of a hybrid process software requirements management*. Las Palmas, Spain.

[21] B. Tanveer, 2017. *Guidelines for utilizing change impact analysis when estimating effort in agile software development*. proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering.

[22] W. Shim and S. Lee, 2017. *An Agile Approach for Managing Requirements to Improve Learning and Adaptability*. Lisbon, Portugal.

[23] J. Arthur and B. James, 2017. *Applying Standard Independent Verification and Validation (IV&V) Techniques within an Agile Framework: Is there a Compatibility Issue?* Montreal, QC, Canada.

[24] B.W.F.V da Silva and S.R.B. Oliveira, 2019. *Challenges of Requirements Management in Agile Methodologies: A Systematic Review*. In: 16<sup>th</sup> CONTECSI.