Rastreabilidade de Requisitos em Metodologias Ágeis: um Estudo Exploratório

Alternative Title: Requirements Traceability in Agile Methodologies: A Exploratory Survey

Gabriela Oliveira da Trindade
Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte (UFRN)
CEP: 59078-970 – Natal, RN – Brasil
g.trindadi@gmail.com

Márcia Lucena
Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte (UFRN)
CEP: 59078-970 – Natal, RN – Brasil
marciaj@dimap.ufrn.br

RESUMO
Metodologias Ágeis estão cada vez mais presentes na indústria. Conhecida como uma metodologia que se adapta facilmente a mudanças, que diminui riscos e proporciona a criação do produto de maneira rápida e segura, possui características como: constante contato com cliente, geração de releases ao final de cada iteração, priorização no que tem mais valor para o cliente, além de "abraçar" facilmente as mudanças solicitadas. Sendo rastreabilidade de Requisitos importante quando pensamos em mudanças, pois facilita atividades como análise de impacto, é possível perceber que esta seria uma atividade bastante útil no processo ágil. No entanto, existem alguns desafios ao tentar aplicar a rastreabilidade em ambientes ágeis. Assim, este trabalho tem o objetivo de realizar uma pesquisa exploratória para levantar os principais problemas relacionados a rastreabilidade em ambientes ágeis, coletando assim informações que ajude no levantamento de requisitos necessários para uma ferramenta que dê suporte a rastreabilidade em ambientes ágeis, e que realize a pré-rastreabilidade, inter-rastreabilidade e pós-rastreabilidade.

Palavras-Chave
Rastreabilidade de Requisitos, Metodologias Ágeis, Gerenciamento de Requisitos.

ABSTRACT
Agile methodologies are increasingly present in the industry. Known as a methodology that adapts easily to changes, which reduces risk and provides product creation quickly and safely, it has features such as: constant contact with the client, generation of releases at the end of each iteration, prioritizing what has more value to the client, in addition to "embrace" easily the changes requested. Being requirements traceability important when we think of changes as it facilitates activities such as impact analysis, it is possible to realize that it would be a very useful activity in the agile process. However, there are some challenges when trying to apply traceability in agile environments. This paper is aimed to perform an exploratory research to raise the main problems related to traceability in agile environments, thus collecting information to assist in raising requirements for a tool that supports traceability in agile environments, and perform pre-traceability inter-tracking and post-traceability.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SBSI 2016, May 17–20, 2016, Florianópolis, Santa Catarina, Brazil.
Copyright SBC 2016.

Categories and Subject Descriptors
D.2.1 Requirements/Specifications (D.3.1)
Methodologies, tools

General Terms
Management, Documentation

Keywords
Requirements Traceability, Agile Methodologies, Requirements Management.

1. INTRODUÇÃO
O processo de desenvolvimento de *software* é composto por etapas que são guiadas com base na metodologia implantada dentro da empresa. Estas metodologias sugerem atividades a serem realizadas durante o desenvolvimento para atingir o objetivo de forma produtiva e eficaz. Metodologias ágeis (MA) de desenvolvimento de software têm se tornado bastante populares, tendo como objetivo a entrega do produto de forma mais rápida e coerente compondo todos os requisitos solicitados [15], [34]. De acordo com [15] os princípios e abordagens das MA são: aumentar a satisfação do cliente, tornar as alterações nos requisitos flexíveis, entregar novas versões frequentemente e aproximar desenvolvedores e clientes.

A priorização na interação entre pessoas ao invés de processos e ferramentas leva a uma falta de documentação de requisitos formal, dificultando a Engenharia de Requisitos (ER). Apesar de MA gerar documentação [47], nem sempre elas são suficientes para a realização da rastreabilidade de requisitos, uma das principais atividades da ER. Como exemplo, podemos citar o código como documentação gerada em MA; entretanto, na maioria das vezes, o código está relacionado apenas com uma história ou tarefa, sem permitir uma visualização de dependências entre as próprias histórias de usuário. Para [14] a falta de documentação adequada pode dificultar a rastreabilidade e consequentemente afetar a gestão de mudança, estimativa e a análise de impacto durante uma alteração.

Muitas são as definições encontradas para rastreabilidade de requisitos (RR). A mais citada entre os trabalhos lidos é a definição de [19], onde para os autores a RR é a capacidade de descrever e seguir a vida de um requisito em ambas as direções, seja do início ao final ou do final ao início de criação do requisito. Já [14] acrescentam a possibilidade de poder analisar as dependências de um requisito.

Sabendo que MA são compostas por ambientes de desenvolvimento dinâmicos, onde existe a participação do cliente dando um *feedback* do sistema e assim havendo uma constante modificação dos requisitos [24], é importante que essas alterações ocorram de forma segura para não invalidar o produto, pois a má elaboração de um requisito pode incapacitar e prejudicar todo o sistema [16]. Contudo, muitas vezes as informações sobre onde uma mudança irá afetar são buscadas no conhecimento da própria equipe e com a documentação mínima exigida pelas MA, o que não garante que uma informação esteja sendo lembrada. Quando a RR é aplicada corretamente ajuda a obter informações essenciais para a realização de uma modificação sem que viole as principais regras do sistema. A rastreabilidade permite que uma análise seja feita sobre o histórico daquele requisito e suas dependências, trazendo vantagens que podem prover a qualidade do sistema.

Apesar das diversas vantagens da RR, a falta de informação sobre seu conceito e objetivos é um fator agravante para a motivação do seu uso [23], causando o desinteresse da própria equipe para seguir práticas que ajudam na rastreabilidade. Podemos pensar em um caso como: Por que documentar informações que ajudam a obter rastreabilidade se um colega de equipe ao meu lado pode me ajudar a tê-las? O grande problema é que essas informações muitas vezes são esquecidas ou perdidas ao longo do tempo, podendo ser ocasionada pela troca de um membro de equipe ou casos em que existem equipes distribuídas [23]. Além disso, [11] questionam a garantia da existência de um profissional que poderá analisar todo o impacto que uma mudança pode causar, sem apoio de recursos que ajude a verificar as dependências existentes em um requisito.

Visando se aprofundar mais sobre os desafios e tendo conhecimento da importância da RR, o presente trabalho busca levantar requisitos para uma ferramenta de gerenciamento de requisitos que permita a rastreabilidade, podendo ser aplicada a um ambiente de desenvolvimento ágil, considerando seus princípios. Para atingir este objetivo, é realizada uma pesquisa exploratória que busca pelos desafios de aplicar a RR em MA, assim como trabalhos que ajudassem no levantamento de requisitos. A pesquisa exploratória é caracterizada por uma abordagem com procedimentos mais flexíveis e consequentemente menos formais. Contudo, neste artigo tentaremos detalhar os critérios adotados para a seleção dos trabalhos. O trabalho está organizado da seguinte forma: a seção 2 se refere a fundamentação teórica, onde está presente as definições e termos utilizados no artigo; a seção 3 apresenta a pesquisa exploratória, mostrando cada uma das etapas realizadas e detalhando os critérios adotados; a seção 4 apresenta uma análise dos requisitos levantados com base na seção 3; e finalmente, a seção 5 que apresenta as conclusões e trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta uma breve explanação sobre metodologias ágeis e definições a respeito de rastreabilidade de requisitos.

2.1 Métodos Ágeis

Os métodos ágeis surgiram a partir de modos de trabalhos diferentes aos conceitos apresentados nas Metodologias Tradicionais [4]. Segundo [24] e [34] uma das mudanças foi a minimização da formalidade e documentação criada no início do desenvolvimento do projeto, tendo o foco maior no código fonte. Com a ideia de permitir que a metodologia pudesse ser usada por todos, ocorreu o Manifesto Ágil, onde foram formulados quatro valores básicos [4], que são: (i) Indivíduos e interações acima de processos e ferramentas; (ii) Software operando acima da documentação abrangente; (iii) Colaboração do cliente acima de negociações contratuais; (iv) Responder às mudanças acima de seguir um plano elaborado.

A partir desses valores vemos que MA priorizam mais o relacionamento entre as pessoas e o produto ao invés do processo [24], fazendo com que a forma de realização das atividades e algumas etapas de metodologias tradicionais fossem modificadas e até extintas [3] [24]. Para [5] a comunicação informal das MA evita a necessidade de documentação e processos de aprovação demorados. Sabendo que MA são adeptas a mudanças, para [24] os agilistas afirmam que os requisitos mudam tão rápido que uma especificação de requisito fica desatualizada logo após ser escrita, desperdiçando esforços; isso faz com que os requisitos sejam elaborados de forma incremental com envolvimento constante do cliente. Além dessa característica citada acima, algumas práticas e comportamentos são bem comuns nas MA, tais como: adaptabilidade às necessidades específicas da equipe; desenvolvimento incremental com ciclos de curtas iterações, priorização do que tem mais valor para o cliente antes de cada iteração, releases frequentes, simplicidade no aprendizado e grande envolvimento do cliente como membro de equipe [6], [42]. Apesar dessas características trazerem grandes vantagens, podemos citar algumas desvantagens, como a eficácia na comunicação com o cliente que depende da sua disponibilidade [5] e o desafio relacionado à rastreabilidade de requisitos [11], que é considerada uma atividade importante no processo de desenvolvimento do software.

2.2 Rastreabilidade de Requisitos

A rastreabilidade de requisitos (RR) está relacionada à qualidade do produto e processo, estando presente em modelos de avaliação como: CMM, CMMI e ISO9001, onde consideram uma atividade essencial para o Gerenciamento de Requisitos [41]. Segundo [44], um requisito é rastreável quando é possível, a partir dele, descobrir quem o sugeriu, o motivo de sua existência, além das dependências. Já [16] e [38] afirmam que a rastreabilidade está associada à capacidade de criar vínculos entre os requisitos e suas fontes e artefatos criados durante o ciclo de vida do desenvolvimento do sistema. Rastrear, através dos artefatos de um ciclo de vida do software, o código, teste e requisitos é relevante para o sucesso de projetos extensos [27]. Algumas das vantagens são sugeridas por [13], [16], [17], [28], [36], [40], [41], onde podemos citar a verificabilidade, que possibilita a análise da implementação de um requisito no sistema e os requisitos que ainda não foram testados; identificação de requisitos e propriedades desnecessárias no sistema, permitindo perceber aqueles requisitos ou propriedades que não colaboram com nenhuma meta do sistema; análise de impacto, ajudando a investigar os efeitos causados por mudanças nos requisitos; correções de defeitos, indicando quais artefatos devem ser analisados para identificar o erro; reusabilidade, ajudando na

reutilização dos artefatos de requisitos; previsão de custos e prazos, permitindo uma análise dos componentes atingidos e assim gerando uma estimativa.

Para [10], [16] e [40] existem habilidades de rastrear que são fundamentais no processo de RR que são: (i) para frente (*forward*), que rastreia um requisito em direção aos seus refinamentos; e (ii) para trás (*backward*): que rastreia um refinamento até sua origem. Além das habilidades acima citadas, algumas classificações de RR foram encontradas na literatura, onde consideraremos 2 delas que são bastante utilizadas e estão presentes em [10], [16], [19], [35], [36] e [40] , que são: (i) Rastreabilidade Pré, Inter e Pós-especificação; e (ii) Rastreabilidade Horizontal e Vertical. A Rastreabilidade pré-especificação está relacionado com os artefatos que formam a base do requisito, assim como os relacionamentos entre os requisitos. Enquanto a rastreabilidade pós-especificação envolve a rastreabilidade entre os requisitos e com os artefatos posteriores que serão incluídos na especificação das próximas atividades do desenvolvimento. A rastreabilidade inter-requisitos: contém o mapeamento das dependências entre requisitos. Enquanto que a rastreabilidade horizontal busca informações das versões e variações dos requisitos ou outros artefatos em uma informada fase do ciclo de vida; a rastreabilidade vertical é realizada entre requisitos e artefatos produzidos durante o ciclo de vida do projeto.

Analisando estas classificações, percebemos que não é uma atividade fácil de ser implantada em um ambiente de MA, pois muitas das práticas adotadas na rastreabilidade são consideradas 'pesadas' [18] [47], além da dependência de uma documentação adequada com informações detalhadas [11] [17] [26]. Entretanto, para [27] a rastreabilidade é fundamental na manutenção do *software*, pois permite uma melhor análise de impacto; sendo assim um importante fator nos ambientes ágeis que procuram sempre 'abraçar' as mudanças. Apesar da sua importância, a aplicação da RR ainda é um desafio em MA [1] [17] [18] [22] [26] [27] [37] [47] [46]. Um trabalho realizado por [23] mostra os motivos pelos quais os agilistas são contra a aplicação da rastreabilidade em MA. Entre eles está a perda de criatividade; crença que toda documentação é contra a filosofia ágil; o fato de não lerem documentos detalhados de especificação; a rotina complexa para garantir de rastreabilidade; perda de tempo na criação causando uma sobrecarga administrativa; além da falta de motivação por não saberem os benefícios que ela possa trazer. Em [23] é sugerido o uso de ferramentas ágeis e automação no processo tornando a rastreabilidade um processo menos 'pesado'.

3. PESQUISA EXPLORATÓRIA

Estudo Exploratório, também conhecido como Pesquisa Exploratória (PE), é classificado como uma Pesquisa de Marketing onde, segundo [31], identifica, coleta, analisa e dissemina informações de forma sistemática e objetiva, ajudando na tomada de decisões relacionadas à identificação e à solução de problemas. Para [32] este tipo de pesquisa é usada para fornecer um maior conhecimento ao pesquisador sobre o tema ou problema abordado. Também segundo [9], a PE é realizada sobre questões de pesquisa que possui pouco ou nenhum estudo anteriormente realizado [9].

Existem algumas classificações de métodos para uma PE [32] e destacamos Levantamento em Fontes Secundárias que é realizada com base em dados coletados ou estudos realizados, podendo ser levantamento do tipo: bibliográfico, documental, estatístico e de pesquisas realizadas. Este trabalho segue esta classificação, pois realiza um levantamento bibliográfico buscando trabalhos feitos sobre o tema abordado que tem como objetivo buscar informações que ajudem na elaboração de uma lista de requisitos para uma ferramenta que permita a RR e possa ser aplicada em um ambiente ágil. Para atingir seu principal objetivo, a PE está dividida em 3 etapas gerais: Planejamento, Coleta de Dados e Análise e Descrição dos Resultados (Figura 1). As próximas seções tratam detalhadamente sobre cada etapa.

3.1 Planejamento de Pesquisa

O planejamento de pesquisa é composto pelas seguintes fases, que são definidas de acordo com [31]: (i) Definição do problema; (ii) Desenvolvimento de uma abordagem e (iii) Formulação e concepção de pesquisa. Na primeira fase o pesquisador deve considerar a finalidade do estudo, as informações relevantes sobre o histórico do problema, quais são as informações necessárias e como elas serão utilizadas na tomada de decisão. A análise de dados secundários pode ajudar nesta definição. O problema irá facilitar o entendimento do objetivo do trabalho. Enquanto na segunda fase é definida a formatação da estrutura objetiva ou teórica, nesta fase teremos questões de pesquisa, além de hipóteses e identificação de características ou fatores que influenciam a concepção de pesquisa. Finalmente a terceira fase busca detectar as fontes relevantes de informações e avaliam os métodos de coleta de dados. Estas 3 fases podem ser encontradas de forma detalhada no link http://1drv.ms/1Q8M68z.

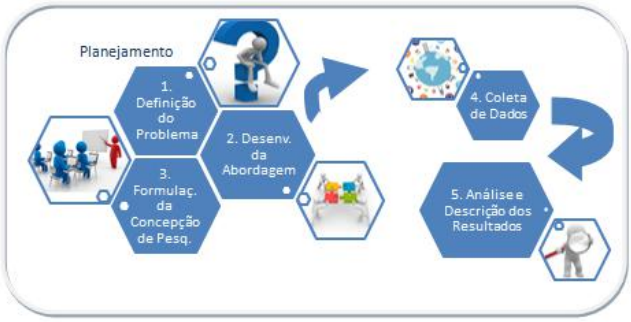

Figura 1. Passos da Pesquisa Exploratória.

3.2 Coleta de Dados

A coleta de dados detalha como os dados foram obtidos para responder o problema [48], diferente da Formulação e Concepção de Pesquisa que detalhou aonde os dados foram buscados. Os dados obtidos nesta pesquisa são secundários, ou seja, provenientes de outros trabalhos que abordam o tema. Os seguintes fatores foram atendidos para a seleção: (i) Presença das palavras-chave citadas no planejamento (ii) Disponibilidade do material, sendo ele aberto; (iii) O idioma do material, podendo ser inglês, português e espanhol; e (iv) O tema principal do trabalho deve se relacionar com as palavras-chave e questões de pesquisa (QP). Primeiramente alguns trabalhos foram selecionados pelo título e resumo, com sua leitura algumas referências presentes também eram analisadas. Dessa forma, foram selecionados àqueles que respondiam as QP.

3.3 Preparação e Análise dos Resultados

Nesta etapa são consideradas as informações dadas na Coleta de Dados e Formulação da Concepção de Pesquisa. Ao final desta seção teremos todas as QP respondidas com base na busca

realizada. É importante lembrar que foram considerados apenas trabalhos que cumprissem os 4 fatores apresentados na Coleta de Dados; portanto, para a seleção, os assuntos de RR e MA deviam estar presentes e relacionados. Este fato foi considerado devido ter percebido que os assuntos tratados de forma separada traziam soluções que interferiam na validação do outro; por exemplo, quando é abordada a RR sem relacionar à MA, geralmente o trabalho apresenta características que na verdade são adotadas em Metodologias Tradicionais, o que tiraria o foco do nosso trabalho. Os trabalhos selecionados estão presentes no link: http://1drv.ms/1Q8M68z, onde existe uma tabela composta pelos Identificadores das Referências, Título, breve Descrição do trabalho, a(s) QP que ele responde, o(s) Autor(es) e Ano da Publicação. Foram selecionados 27 produções, para cada produção que possuir uma QP respondida, ela é comentada na resposta dada. Por motivos de espaço, neste artigo estão presentes apenas as questões de pesquisa QP1 e QP4, contudo será possível verificar as resposta da QP2 e QP3 no link junto com os trabalhos selecionados, apresentado acima.

QP1. Quais problemas/desafios estão relacionados à rastreabilidade de requisitos no contexto de desenvolvimento ágil?

Segundo [14] a rastreabilidade foi projetada para o desenvolvimento de software tradicional, devido as suas características. Com isso foi fácil perceber que seria um desafio propor RR em MA. Alguns autores abordam esses desafios. Em [1] os principais desafios estão na garantia da rastreabilidade a partir de Especificação de Requisito, História de Usuário e Artefatos de Projeto, uma vez que as ferramentas que geram estes artefatos não possuem um mesmo padrão. Além disso, o fato de manter a rastreabilidade e torná-la sempre consistente, já que muitos dos artefatos em MA são criados por diferentes pessoas com diferentes papéis. Uma entrevista é realizada por [22], em que a maioria dos entrevistados relatam gastos com recursos para aplicar a rastreabilidade, sendo eles: tempo, funcionários e software. Também é comentado sobre a experiência necessária da equipe para saber o nível de rastreabilidade que deve ser adotado ou até mesmo como utilizar as ferramentas. Além desses problemas, o trabalho cita a motivação que deve ser dada a equipe sobre a importância da RR, cooperação que deve existir entre os membros para manter a documentação atualizada e também sobre a dificuldade de analisar dependências em um projeto grande que provavelmente possuiria diversos *links*. Da mesma forma, [7] cita problemas que, apesar de serem relacionados a métodos tradicionais, também se encaixariam no paradigma ágil, que seriam: tempo dedicado, foco em documentação e manutenção da matriz de rastreabilidade. Para [18] a RR depende de muitos pressupostos que não são práticas de ambientes ágeis; como exemplo o autor cita a da matriz de rastreabilidade, que é a criação de um documento complementar, além da dependência de documentos formais de requisitos [25] [26]. Já [46] afirmam de uma forma mais geral que é um desafio definir um esquema que atenda RR e MA, pois a natureza dos artefatos gerados em um ambiente ágil torna difícil para modelos estáticos e centrados em documentos, implicando na execução da atividade de rastreabilidade [27].

Outros autores citam o uso da prática de desenvolvimento ágil *Test Driven Development* (TDD). [30] afirmam a possibilidade de usar o TDD para a rastreabilidade. Entretanto, [30] e [20] dizem que a refatoração ocorrida no TDD pode ser um grande desafio, pois com ela podem surgir novas dependências ou até mesmo desaparecer vínculos entre testes e código fonte. Isto pode gerar uma matriz inconsistente com os dados e afetar ainda mais quando usada na refatoração [20]. Segundo [37] desafios como relacionar código com testes que testam muitas classes em diversas situações, assim como código com testes de unidade que contêm métodos que executam testes em outros objetos. Para [47] "é difícil integrar e aplicar elementos de rastreabilidade para MA de acordo como está firmado pelo Manifesto Ágil".

QP4. Quais são as principais funcionalidades que dão suporte a rastreabilidade de requisitos em uma ferramenta para gerenciamento de requisitos no contexto de metodologia ágil?

Uma entrevista realizada por [23] traz resultados que de forma indireta sugerem requisitos como: não deixar o processo pesado, as informações de rastreabilidade serem de fácil acesso, consulta e manipulação. Além desses requisitos, [23] também apresenta informações como: possibilidade de voltar e ver de onde um requisito surgiu, de criar um *feedback* e informar ao interessado quando o requisito é implementado; lista de pendências, assim como a descrição do problema inicial; possibilitar o vínculo de requisitos e história, assim como sua remoção ou atualização; possibilitar o vínculo de requisito para requisito, assim como requisito para código; possibilitar o vínculo de requisito para versão; poder validar os requisitos testados; permitir a avaliação do usuário para a versão; permitir visualizar no que os outros estão trabalhando; entre outros. [15] completam com as seguintes características: tornar o processo de RR implícito e minimamente invasivo, realizar *links* entre histórias de usuário e casos de testes, assim como arquitetura. Em [2] é elaborado um modelo e para este é apresentado uma lista de casos de uso de uma ferramenta que suporte o seu modelo, sendo composta por: Login, Gerenciar Projeto, Gerenciar Fase, Gerenciar História de Usuário, Gerenciar Requisito, Gerenciar Associação, Gerenciar Recurso de Segurança, Gerenciar Recurso de Desempenho, Gerenciar Caso de Teste e Gerenciar Desenvolvimento.

Já no trabalho [10], casos de uso como: Escolher Escopo; Definir Valor; Escrever, Estimar e Aceitar Stories; Definir Requisitos; Cadastrar Equipe, Projeto, Realeases, Iterações, Meetings, Eventos e Recursos, são apresentados. [10] ainda acrescenta uma consulta ao log de rastreabilidade que traz informações de quem realizou certa operação, quando foi realizada, o que foi feito, um link para o estado atual da história e uma 'fotografia' da história no momento anterior ao que a operação foi realizada, além de dá suporte a rastreabilidade entre histórias ou recursos (PDF ou PNG) através de conectores como *<story>* e *<resource>*, respectivamente. [14] mostram a restrição de acesso a certas funções, dependendo do papel do ator.

Funcionalidades como visualizações de dados de rastreabilidade são acessadas apenas por partes interessadas. Mostra ainda os seguintes requisitos: informações dos níveis de granularidade dos artefatos para ligações, pesos para indicar o nível de dependência entre dois artefatos traçados, métricas de rastreabilidade, ou as diretrizes para indicar os artefatos corretos para vincular. A ferramenta apresentada por [15] é baseada na metodologia *Scrum*, contendo um *product backlog* que apresenta todas as histórias de usuário que o compõe. Esta ferramenta permite identificar quais histórias foram utilizadas para a construção de uma funcionalidade e, se cadastrados pelos desenvolvedores, quais arquivos fontes estão associados aos requisitos. Também permite verificar se o relacionamento entre os requisitos é forte (com

grande impacto) ou fraco através de uma matriz de rastreabilidade ou através da própria funcionalidade de visualizar o requisito. Já [39] optam pelo sistema de gestão *Kanban*, onde implementam uma ferramenta *web* que possibilita ao usuário acompanhar o progresso e monitoramento dos requisitos, a demonstração dos requisitos que o sistema atende e que estes foram aplicados corretamente, avaliar e validar os requisitos, coletar informação histórica para a manutenção do produto através do rastreamento – desde sua origem a seu desenvolvimento, cadastrar projetos e colaboradores e registrar de requisitos. A partir de documentos do *product backlog* e *Sprint backlog*, [12] criam *links* de rastreabilidade. Em seu trabalho uma ferramenta é desenvolvida com a funcionalidade de buscar documentos que a ferramenta possui a partir de palavras-chave informadas pelo usuário e campo que se deseja relacionar com a palavra, trazendo informações como: ID do requisito, descrição, projeto e teste. [22] realiza uma pesquisa em que muitas empresas afirmam preferir mecanismos de rastreabilidade que sejam de fácil aprendizado, pois, segundo ele, ferramentas podem ser complexas ao tentar cobrir muitas funcionalidades.

Uma abordagem baseada em ferramenta é sugerida por [27], que implementa um *plug-in* para o Eclipse, onde o mesmo permite a gravação de artefatos de desenvolvimento de software como especificações de requisitos. Ela grava e compartilha informações reunidas durante intercâmbios com os clientes, tornando este artefato parte da especificação e conectando-o através de uma *tag*. Além disso é possível realizar uma avaliação de impacto das mudanças através de ligações capturadas, possibilitando a pós e pré-rastreabilidade. [33] apresentam a ferramenta Controlle, que possui funcionalidades como: manter a rastreabilidade na própria tela de cadastro do artefato ou de modo visual através de um grafo, relacionar um artefato a qualquer outro (requisito, protótipo de tela, arquivos de código, atas de reunião, casos de uso, tabela, entre outros) e definir o tipo de relacionamento e direção são listadas. É possível ainda gerar um grafo e nele diferenciar a rastreabilidade horizontal e vertical, além de permitir visualizar informações detalhadas e navegar entre os artefatos.

Em seu trabalho, [45] falam da importância de MA possuirem rastreabilidade. Com isto, apresentam uma ferramenta que não foi desenvolvida para ambiente ágil, mas tentam ajustá-la para dá suporte a MA. Então citam requisitos como: definições gerais do escopo, análise de riscos, requisitos gerais de segurança, requisitos funcionais e não-funcionais, árvore de rastreabilidade, tabela de atributos com informações como: identificador, autor, data, descrição e suposições.

Uma ferramenta chamada RETRO é desenvolvida por [21]. Seu foco não está em MA, contudo ela realiza a rastreabilidade de maneira automatizada. Esta ferramenta possibilita o usuário poder adicionar ou remover *links* que foram gerados automaticamente, além de realizar consultas de *links* através de informações como um peso, por exemplo. Já o *StoryManager* foi sugerida por [25], sendo uma ferramenta aplicada ao ambiente *eXtreme Programming* (XP) que gerencia histórias de usuário de tarefas. Algumas características para ferramentas aplicadas a MA são citadas por [25], onde podemos citar: não forçar os procedimentos detalhados, fornecendo o suficiente e básico; permitir que o processo continue ágil; integrar o suporte da ferramenta de gerenciamento de requisitos em ambiente de desenvolvimento global do projeto. Todos esses trabalhos fazem perceber que muitos autores estão buscando soluções através de sugestões de ferramentas. Essa foi uma solução pensada inicialmente, onde a PE ajudou a adquirir mais conhecimento sobre RR em MA e assim apresentar requisitos que seja consistente com os dois temas.

4. ANÁLISE DOS REQUISITOS

Esta seção apresenta a lista de Requisitos Funcionais (RF) e Não-funcionais (RNF) para uma ferramenta de gerenciamento de requisitos que suporte a rastreabilidade e práticas da MA. Esses requisitos foram levantados com base nos trabalhos selecionados a partir da PE, onde alguns já apresentavam a ideia de uma ferramenta que auxiliasse a RR em MA, contudo muitas delas focavam em uma metodologia específica. Antes da listagem dos requisitos, é necessário apresentar seu conceito, assim como sua classificação. Para [43] requisitos são descrições de serviços e funcionalidades que devem estar presentes no sistema, assim como suas restrições operacionais. Essas funcionalidades são elicitadas de acordo com as necessidades do cliente, objetivando sua satisfação. De acordo com [43], os requisitos funcionais (RF) declaram funções que o sistema deve fornecer, como o sistema deve reagir a entradas especificas e como deve ser o seu comportamento em determinadas situações. Enquanto que os requisitos não funcionais (RNF) são caracterizados por restrições sobre as funções oferecidas pelo sistema. A Tabela 1 apresenta uma lista de RFs e RNFs, sendo classificados em básicos – ou seja, aqueles que farão o sistema funcionar de acordo com MA – e os diretamente ligados à RR.

Tabela 1. Requisitos Funcionais

ID	Requisitos Funcionais Básicos
RF1	O sistema deve possibilitar a listagem, cadastramento, alteração e exclusão de usuários.
RF2	O sistema deve possibilitar ao usuário a listagem, criação, alteração e exclusão de um projeto.
RF3	O sistema deve possibilitar ao usuário a listagem, criação, alteração e exclusão de uma iteração.
RF4	O sistema deve possibilitar ao usuário a criação, alteração e exclusão de histórias de usuário.
RF5	O sistema deve possibilitar a criação, alteração e exclusão de tarefas.
RF6	O sistema deve listar todos os projetos que o usuário logado participa.
RF7	O sistema deve permitir o usuário listar/consultar todas as histórias, tarefas e testes criadas dado um projeto.
RF8	O sistema deve possibilitar ao usuário a criação, alteração e exclusão de teste.
RF9	O sistema deve possibilitar ao usuário estimar histórias de usuário.
RF10	O sistema deve possibilitar ao usuário mudar a situação das histórias de usuário, tarefas e testes.
RF11	O sistema deve possibilitar ao usuário priorizar histórias de usuário.
RF12	O sistema deve permitir o usuário fazer comentários nas histórias, tarefas e testes.
RF13	O sistema deve se integrar a uma ferramenta de gerenciamento de versões.
ID	**Requisitos Funcionais para a Rastreabilidade**
RF14	O sistema deve permitir gerar um grafo de rastreabilidade entre as tarefas, testes e/ou histórias de usuário.
RF15	O sistema deve possibilitar cadastrar, alterar e excluir uma reunião de áudio ou vídeo em uma determinada

	tarefa relacionada a ela.
RF16	O sistema deve permitir ao usuário, no momento de cadastro ou alteração de histórias criar/remover o vínculo com outras histórias, tarefas e/ou testes.
RF17	O sistema deve permitir ao usuário, no momento de cadastro ou alteração de tarefas criar/remover o vínculo com outras histórias, tarefas e/ou testes.
RF18	O sistema deve permitir ao usuário, no momento de cadastro ou alteração de testes criar/remover o vínculo com outras histórias, tarefas e/ou testes.
RF19	O sistema deve gerar automaticamente sugestões de links de rastreabilidade, no momento do cadastro ou alteração, a partir de palavras-chaves escritas nas histórias e artefatos ou tarefas.
RF20	O sistema deve possibilitar ao usuário relacionar histórias a iteração.
RF21	O sistema deve permitir o usuário relacionar/remover relação entre tarefas e arquitetura.
RF22	O sistema deve permitir o usuário relacionar/remover relação entre histórias de usuário e código-fonte, gerados automaticamente a partir de commits onde é informado o número da tarefa.
RF23	O sistema deve permitir o usuário relacionar/remover relação entre tarefas e documentos gerados a partir dela (protótipos, ata de reunião,...).
RF24	O sistema deve permitir o usuário relacionar/remover relação entre histórias de usuário e documentos gerados a partir dela (protótipos, ata de reunião,...).
RF25	O sistema deve permitir o usuário informar um peso (indicando a força da dependência) para o rastro que está cadastrando.
RF26	O sistema deve permitir o usuário informar as diretrizes para indicar os artefatos a vincular.
RF27	O sistema deve permitir a análise de impacto, dado uma história de usuário ou tarefa.
RF28	O sistema deve possibilitar ao usuário a visualização de histórias, trazendo um log de rastreabilidade.
RF29	O sistema deve permitir relacionar usuários e artefatos.

Tabela 2. Requisitos Não-funcionais

ID	Requisitos Não-funcionais Básicos
RNF1	Tornar o processo implícito, minimamente invasivo e leve.
RNF2	Fácil usabilidade.
RNF3	Não forçar procedimentos detalhados.
RNF4	O sistema deve suportar trabalho colaborativo.
ID	**Requisitos Não-funcionais para a Rastreabilidade**
RNF5	Facilitar o acesso, manipulação e consulta às informações de rastreabilidade.
RNF6	Manter a rastreabilidade com os dados informados na ferramenta.

Do RF1 ao RF13 temos requisitos que buscam inserir informações sobre o projeto a ser gerenciado, atendendo às características de metodologias ágeis. Podemos ver funcionalidades como manter usuário, que ajuda no controle de permissões e o registro de operações; manter projeto, onde será possível tratar projetos de diferentes clientes de forma separada; manter iterações, possibilitando seguir uma prática comum em MA, que é gerar um produto ao final de cada iteração; manter histórias de usuário, tarefas e testes, que compõem a criação de uma funcionalidade. Além disso, temos também a priorização e estimativa de histórias, mudança de situação de histórias, tarefas e testes que são práticas bastante comuns em MA. Os requisitos relacionados à RR estão definidos do RF14 ao RF29, onde a partir das informações registradas pelos requisitos básicos, será possível realizar atividades da criação e geração de rastreabilidade. Em específico, cito o requisito RF14 que permite gerar um grafo de rastreabilidade entre tarefas, testes e/ou histórias de usuário. Duas formas bem comuns para a visualização de dependências são matriz de rastreabilidade e grafos. A escolha do grafo foi realizada por ser adequado para a navegação entre rastros [8] [29], enquanto a matriz de rastreabilidade torna a leitura difícil quando gerada em projetos com muitos requisitos [22]. De forma similar a RF27 também gera um grafo, contudo é preciso informar uma história, tarefa ou teste que se quer analisar e assim o grafo é gerado a partir dele, mostrando também a força da dependência entre os artefatos presentes nele (RF25). Entre os RF ligados a rastreabilidade, alguns permitem que sejam criados vínculos de rastreabilidade (RF15, RF16, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF24, RF29), dando destaque ao RF15 que se preocupa em documentar e vincular reuniões comuns nas MA onde as informações muitas vezes não são registradas, ficando apenas na memória de membros da equipe [1], e também ao RF29 que relaciona usuário ao artefato, tornando possível verificar o responsável por uma tarefa, por exemplo. Muitos dos requisitos permitem não apenas a criação do vínculo, mas também a sua remoção (com exceção do RF29), pois devemos considerar que algumas informações podem ter sido inseridas de forma errada, como também o fato de um vínculo deixar de existir após uma modificação [45]. O requisito RF25 tem o objetivo de indicar a força da dependência entre dois artefatos, sendo ela forte ou fraca, onde a dependência forte é aquela que tem grande impacto no artefato que está sendo relacionada, enquanto a fraca tem menos impacto [15]. Essas informações são mostradas no grafo gerado a partir de uma análise de impacto (RF27), através de uma cor determinada na aresta que liga os dois artefatos. Enquanto que o RF26 possibilita ao usuário definir as diretrizes para indicar artefatos a vincular [14]. Isto ajuda na determinação das práticas que devem ser seguidas, permitindo que a ferramenta se comporte com o que foi definido nas diretrizes, por exemplo, se vínculo entre histórias e teste não for selecionado, o sistema não terá tal funcionalidade. O RF28 traz uma visualização geral das informações contidas em histórias de usuário, gerando um *log* de rastreabilidade que guarda informações como quem criou a história, as tarefas que a compõe com as informações de quem e quando realizou operações em cima dela, o que foi modificado, os artefatos que estão ligados a ela e se essa ligação é forte ou fraca.

Além dos RF, também foram listados os RNF. Entre estes temos do RNF1 ao RNF4, onde é composto pelos requisitos não-funcionais relacionados às MA, permitindo que o processo continue "leve" (RNF1), não forçando procedimentos detalhados (RNF3); o sistema seja de fácil usabilidade (RNF3), tornando-o ágil; além de permitir o trabalho colaborativo (RNF4), uma característica forte das MA. Já o RNF5 e RNF6 compõem requisitos não-funcionais relacionados à RR, que permitem ser uma funcionalidade de fácil acesso e manipulação, assim como garantir que a RR ser consistentes aos dados informados na ferramenta, respectivamente. A Tabela 3 tem o objetivo de

mostrar quais os requisitos funcionais que atendem ao tipo de rastreabilidade.

Tabela 3 classificação de rastreabilidade e requisitos

Classificação	Requisitos
Pré-rastreabilidade	RF15, RF23, RF24 e RF29
Pós- rastreabilidade	RF15, RF16, RF17, RF18, RF21, RF22, RF23, RF24 e RF29
Inter-rastreabilidade	RF14, RF27, RF28 e RF29
Rastreabilidade Horizontal	RF20 e RF28
Rastreabilidade Vertical	RF16, RF17, RF18, RF21, RF22, RF23, RF24, RF28, RF29

5. CONSIDERAÇÕES FINAIS

A PE fez perceber que a RR não é uma prática comum em ambientes de desenvolvimento ágil. Isto devido as práticas adotadas em MA e atividades necessárias para desenvolver a RR, que são consideradas 'pesadas'. Apesar disso, viu os benefícios que a rastreabilidade poderia trazer em ambientes ágeis.

Percebendo que a rastreabilidade é uma atividade trabalhosa e a importância em realiza-la com o uso de uma ferramenta [43], foram realizadas buscas na PE que abordassem o uso de uma ferramenta para auxiliar a RR em ambientes que adotam MA. Com isso, trabalhos que possuíssem essa sugestão foram encontrados; entretanto, nem sempre eles contemplavam uma combinação da pré-rastreabilidade, inter-rastreabilidade, pós-rastreabilidade, rastreabilidade vertical e horizontal.

Trabalhos como [10], [15] e [39] apresentaram ferramentas dentro do assunto abordado, trazendo alguns requisitos levantados neste artigo. Contudo, tais ferramentas focavam em uma MA em específico, restringindo sua aplicação a outros projetos com metodologias diferentes. Isto difere do presente artigo, que visa apenas atender os valores das MA.

É importante destacar o trabalho realizado por [23], sendo ele o trabalho mais próximo a este. [23] realizou um levantamento de padrões a serem seguidos para contemplar a RR em MA. Mas a grande diferença quando comparado a este, foi a metodologia utilizada – enquanto utilizamos a PE, [23] realiza uma entrevista, onde a maioria eram gerentes, para o levantamento de padrões.

Nossa grande contribuição está na elaboração da lista de requisitos necessários para uma ferramenta de gerenciamento de requisitos com foco em rastreabilidade e desenvolvimento ágil. Destaca-se a preocupação em atender os valores adotados na MA, além das 5 classificações abordadas de rastreabilidade. Além disso, podemos citar a pesquisa, que trouxe trabalhos relevantes com o objetivo de responder as questões e ajudar no desenvolvimento do artigo.

Como trabalhos futuros ficam o desenvolvimento da ferramenta que contenha os requisitos listados, além de uma comparação entre a ferramenta a ser desenvolvida e aquelas apresentadas com a PE.

6. REFERÊNCIAS

[1] Antonino, P.O., Keuler, T., Germann, N., and Cronauer, B. 2014. A non-invasive approach to trace architecture design, requirements specification and agile artifacts. In *Software Engineering Conference (ASWEC), 2014 23rd Australian* IEEE, p. 220-229.

[2] Arbain, F.B.A., Ghani, I., Kadir, W.M.N.W. 2014. Agile non functional requiremnents (NFR) traceability metamodel. In *Software Engineering Conference (MySEC), 2014 8th Malaysian* IEEE, p. 228-233.

[3] Bassi Filho, D.L. 2008. *Experiências com desenvolvimento ágil*, Dissertação de Mestrado, Universidade de São Paulo.

[4] Beck, K. et al. 2001. Manisfesto Ágil, Manifesto para o Desenvolvimento Ágil de Software. Disponível em http://www.manifestoagil.com.br/index.html. Último acesso: Fevereito, 2016.

[5] Cao, L., and Ramesh, B. 2008. Agile requirements engineering practices: An empirical study. *Software, IEEE, v. 25*, n. 1, p. 60-67.

[6] Ciscon, L.A. 2009. *Um Estudo E Uma Ferramenta De Gerência De Projetos Com Desenvolvimento Ágil De Software.* Dissertação De Mestrado. Universidade Federal De Minas Gerais.

[7] Cleland-Huang, J. 2006. Just enough requirements traceability. In *Computer Software and Applications Conference, 2006. COMPSAC'06. 30th Annual International* IEEE, p. 41-42.

[8] Cleland-Huang, J., Gotel, O.C.Z., Huffman Hayes, J., Mader, P., and Zisman, A. 2014. Software traceability: trends and future directions. *In: Proceedings of the on Future of Software Engineering.* ACM. p. 55-69.

[9] Collis, J., and Hussey, R. 2005. *Pesquisa em administração: um guia prático para alunos de graduação e pós-graduação*. 2 ed. Porto Alegre: Bookman.

[10] Dall'Oglio, P. 2006 . Uma Ferramenta para Gerenciamento de Requisitos em Projetos Baseados em Extreme Programming.. Monografia. Universidade do Vale do Rio dos Sinos.

[11] De Rezende Alves, S., and Alves, A.L., Alves–Especialista, A. L. Engenharia De Requisitos Em Métodologias Ágeis.

[12] Duraisamy, G., and Atan, R. 2013. Requirement Traceability Matrix through Documentation for Scrum Methodology. *Journal of Theoretical & Applied Information Technology*, v. 52, n. 2, p. 154-159.

[13] Egyed, A., and Grünbacher, P. 2005. Supporting software understanding with automated requirements traceability. *International Journal of Software Engineering and Knowledge Engineering*, v. 15, n. 05, p. 783-810.

[14] Espinoza, A., and Garbajosa, J. 2011. A study to support agile methods more effectively through traceability. *Innovations in Systems and Software Engineering*, v. 7, n. 1, p. 53-69.

[15] Franceschi, R.A., and Duarte, A.M.D. 2011. Uma abordagem para gerência de requisitos integrada com práticas ágeis de gerência de projetos.

[16] Genvigir, E.C. 2009. *Um modelo para rastreabilidade de requisitos de software baseado em generalização de elos e atributos*. Doctoral Thesis. São José dos Campos: Instituto Nacional de Pesquisas Espaciais.

[17] Ghazarian, A. 2008. Traceability patterns: an approach to requirement-component traceability in agile software

development. In *Proceedings of the 8th conference on Applied computer scince*. World Scientific and Engineering Academy and Society (WSEAS). p. 236-241.

[18] Ghazarian, A. 2010. Reliability in agile software engineering: A dilemma. Technical report, IEEE Reliability Society, East Lansing, Michigan, 2011. 1.

[19] Gotel, O.C.Z., and Finkelstein, A.C.W. 1994. An analysis of the requirements traceability problem. *In Requirements Engineering, 1994, Proceedings of the First International Conference on IEEE*, 94-101.

[20] Hayes, J.H., Dekhtyar, A., and Janzen, D.S. 2009. Towards traceable test-driven development. In *Traceability in Emerging Forms of Software Engineering, 2009. TEFSE'09. ICSE Workshop on* IEEE, p. 26-30.

[21] Hayes, J.H., Dekhtyar, A., Sundaram, S.K., Holbrook, E.A., Vadlamudi, S., and April, A. 2007. REquirements TRacing On target (RETRO): improving software maintenance through traceability recovery. *Innovations in Systems and Software Engineering*, v. 3, n. 3, p. 193-202.

[22] Hoang Duc, V. 2013. Traceability in Agile software projects. Master Thesis. University of Gothenburg, Chalmers University of Technology.

[23] Jacobsson, M. 2009. *Implementing traceability in agile software development*. Master Thesis. Department of Computer Science, Lund University.

[24] Jaqueira, A.D.O.P. 2013. *Uso de modelos i* para enriquecer requisitos em métodos ágeis*. Master Thesis. Departamento de Informática e Matemática Aplicada, Universidade Federal do Rio Grande do Norte.

[25] Kääriäinen, J., Koskela, J., Takalo, J., Abrahamsson, P., and Kolehmainen, K. 2003. Supporting requirements engineering in extreme programming: managing user stories. In *Proc. 16th International Conference on Software & Systems Engineering and their Applications (ICSSEA 2003)*.

[26] Khan, H.M., and Arshad, I. 2012. Test-lists Utilization in Test Driven Development: The Role of test-lists in Requirements Traceability. Master Thesis. School of Engineering at Blekinge Institute of Technology.

[27] Lee, C., Guadagno, L., and Jia, X. 2003. An agile approach to capturing requirements and traceability. In *Proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE 2003). Citeseer.*

[28] Letelier, P. 2002. A framework for requirements traceability in UML-based projects. In *Proceedings of the 1st International Workshop on Traceability in Emerging Forms of Software Engineering*. p. 30-41.

[29] Li, Y., and Maalej, W. 2012. Which traceability visualization is suitable in this context? a comparative study. *In: Requirements Engineering: Foundation for Software Quality.* Springer Berlin Heidelberg, *2012*. p. 194-210.

[30] Lucia, A.D., and Qusef, A. 2010. Requirements engineering in agile software development. *Journal of Emerging Technologies in Web Intelligence*, v. 2, n. 3, p. 212-220.

[31] Malhotra, N. K. 2006. Pesquisa de Marketing. Porto Alegre,Bookman.

[32] Mattar, F., and Motta, S. 2014. *Pesquisa de Marketing, 7ª Edição*. Elsevier Brasil.

[33] Nascimento, F., Texeira, M., Thiry, M., and Zoucas, A. 2011. Controlle: Ferramenta de Apoio à Gerência de Requisitos.WANPS, 2011.

[34] Paetsch, F., Eberlein, A., and Maurer, F. 2003. Requirements engineering and agile software development. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2003). Proceedings. Twelfth IEEE International Workshops on. IEEE*, 2003. p. 308-313.

[35] Pereira, J.R.R. 2011. *Desenvolvimento de um software para métricas em rastreabilidade de requisitos de software.* Proposta de Trabalho de Diplomação. Universidade Tcnológica Federal do Paraná, Cornélio Procópio.

[36] Pohl, K., and Rupp, C. 2011. *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam-Foundation Level-IREB Compliant.*Rocky Nook, Inc.

[37] Qusef, A. 2013. Test-to-code traceability: Why and how? In *Applied Electrical Engineering and Computing Technologies (AEECT), 2013 IEEE Jordan Conference on* IEEE. p. 1-8.

[38] Ramesh, B., and Jarke, M. 2001. Toward reference models for requirements traceability. *Software Engineering, IEEE Transactions on* 27, 58-93.

[39] Reyes, J.M.H., and Ramírez, C.E.. Sistema de control de trazabilidad de requerimientos en ambientes ágiles de desarrollo de software.

[40] Sá Neto, E. L. 2013. Rastreabilidade de Requisitos. Monografia da disciplina de Engenharia de Requisitos. Universidade Federal de Pernambuco.

[41] Sayão, M., and Do Prado Leite, J.C.S. 2005. Rastreabilidade de requisitos. *RITA*, v. 13, n. 1, p. 57-86, 2005.

[42] Sillitti, A., and Succi, G. 2005. Requirements engineering for agile methods. In *Engineering and Managing Software Requirements*. Springer Berlin Heidelberg, p. 309-326.

[43] Sommerville, I. 2010. Software Engineering. 9th edition – New York- Addison Wesley.

[44] Sommerville, I., and Kotonya, G. 1998. *Requirements Engineering: Processes and Techniques*. John Wiley and Sons, Inc.

[45] Stålhane, T., Katta, V., and Myklebust, T. 2014. Change Impact Analysis in Agile Development. *EHPG Røros*.

[46] Taromirad, M, and Paige, R.F. 2012. Agile requirements traceability using domain-specific modelling languages. In *Proceedings of the 2012 Extreme Modeling Workshop* ACM, p. 45-50.

[47] Urquiza-Yllescas, J.F. and Martínez-Martínez, A. 2011. Trying to Link Traceability Elements in a General Agile Model Life Cycle.

[48] Vergara, S.C. 2000. Começando a definir a metodologia. *Projetos e relatórios de pesquisa em administração, v. 3*, p. 46-53.