A Engenharia de Requisitos nos métodos ágeis: uma revisão sistemática da literatura

Bárbara Silveira Fraga
Av. Afonso Vaz de Melo, 1200
Belo Horizonte – MG - Brasil
barbarasilveiraf@gmail.com

Marcelo Werneck Barbosa
Av. Afonso Vaz de Melo, 1200
Belo Horizonte – MG - Brasil
mwerneck@pucminas.br

RESUMO
Por não possuírem características prescritivas, não há, nos métodos ágeis, uma definição formal sobre como as atividades de Engenharia de Requisitos (ER) devem ser executadas. Isso faz com que haja uma variedade de formas de se definir como serão levantados, priorizados, especificados e validados os requisitos. Assim, torna-se necessário compreender como as pesquisas em ER ágil têm caracterizado estas atividades. O objetivo deste trabalho foi, por meio de uma revisão sistemática da literatura, identificar as práticas e técnicas utilizadas para cada processo da ER em projetos ágeis. O trabalho identificou ainda importantes desafios e lições aprendidas que devem direcionar as evoluções nesta área.

Palavras-Chave
Engenharia de Requisitos, Métodos Ágeis e Revisão Sistemática da Literatura,

ABSTRACT
Since agile methods do not have prescriptive characteristics, there is not a formal definition on how Requirements Engineering (RE) activities should be executed. This situation leads to a variety of different forms of defining how requirements are elicited, prioritized, documented and validated. So, it is necessary to comprehend how research on RE have characterized such activities. By using a systematic review approach, the objective of this study was to identify practices and techniques that have been used for each RE process in agile projects. The study also identified challenges and lessons learned that should direct improvements in this area.

CCS Concepts
• Software and its engineering□ Software creation and management□ Software development process management □ Agile software development • Software and **its** engineering □ Software creation and management □ Designing Software □ Requirements Analysis**.**

Keywords
Requirements Engineering; Agile Methods; Systematic Literature Review.

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 2017, June 5th–8th, 2017, Lavras, Minas Gerais, Brazil.
Copyright SBC 2017.

1. INTRODUÇÃO
Os métodos ágeis estão cada vez mais populares nas organizações, uma vez que eles oferecem uma maior produtividade, um custo menor, melhor qualidade e satisfação do cliente [32][38]. Como estes são objetivos comuns a muitas organizações, muitas empresas têm adotado estes métodos. Entretanto, apesar de trazerem benefícios, a mudança do desenvolvimento tradicional para o ágil não é simples e nem rápida. Quando ocorre a transformação de um processo de desenvolvimento tradicional para o ágil, mudanças técnicas, gerenciais, pessoais e culturais são necessárias, trazendo várias barreiras a serem vencidas [11].

Um dos fatores de sucesso dos métodos ágeis é a insatisfação causada pela burocracia nos métodos tradicionais [35]. Muitas vezes essa burocracia está relacionada à extensa documentação, característica comum dos métodos tradicionais. Enquanto os métodos tradicionais enfatizam o planejamento e o projeto, os métodos ágeis enfatizam a implementação do software. Outra característica importante dos métodos ágeis é a comunicação face a face em oposição à criação de uma documentação extensa, quando julgada não necessária [34]. A ausência de definição formal das atividades e produtos relacionados à Engenharia de Requisitos (ER) faz com que as organizações definam estas atividades de diversas maneiras, sendo necessário compreender como tais atividades têm sido realizadas e como esta definição tem impactado os resultados dos projetos. Outro problema associado a se ter uma documentação extensa é que as mudanças no desenvolvimento de software são constantes e é exigido que o software atenda rapidamente a elas. Isso se torna um desafio para desenvolvimento de software e em alguns casos a documentação e os requisitos se tornam obsoletos antes mesmo do término do projeto [19]. No estudo feito por Mendes e outros [31] foram analisadas 132 tarefas de manutenção e evolução em projetos ágeis. Em 65 dessas, foram notados débitos de documentação. Isso ocasionou um esforço 47% maior do que o estimado no desenvolvimento do projeto e um custo extra de 48%. Assim, orientam este trabalho as seguintes perguntas:

- P1 – Quais práticas de levantamento e detalhamento de requisitos as organizações têm seguido?
- P2 – Quais benefícios as empresas têm alcançado com o uso de tais práticas e quais são os desafios a serem vencidos?
- P3 - Como pode ser caracterizada a pesquisa em Engenharia de Requisitos aplicada a métodos ágeis?

Com o objetivo de elaborar um panorama do uso e pesquisa em requisitos em métodos ágeis, este trabalho realizou uma revisão sistemática no contexto de aplicação da Engenharia de Requisitos em métodos ágeis. O presente artigo está subdividido nas seguintes seções: (2) Referencial Teórico, (3) Metodologia, (4) Resultados, (5) Discussões e Conclusões.

2. REFERENCIAL TEÓRICO

2.1 Métodos Ágeis

A abordagem dos métodos ágeis foi definida pelo Manifesto Ágil, que valoriza os indivíduos e interações ao invés de processos e ferramentas, software funcionando mais que a documentação, colaboração do cliente mais que negociação de contratos e responder a mudanças mais do que seguir um plano [29]. Enquanto os métodos tradicionais de desenvolvimento de software se preocupam com uma documentação abrangente e completa, os métodos ágeis têm a tendência de se referir ao código fonte como único artefato de documentação [35].

O objetivo dos métodos ágeis é solucionar o problema de criação de um software com alta qualidade em tempo hábil frente às constantes mudanças de requisitos no ambiente de negócio. Além disso, ele permite um processo adaptativo, no qual a equipe de desenvolvimento consegue lidar com as mudanças de requisitos, mesmo em um estágio avançado de desenvolvimento [26]. Além disso, quando a equipe se conhece, uma boa comunicação e um bom entrosamento os resultados podem ser ainda melhores. Essa comunicação está muito presente nas reuniões diárias entre o time e o cliente, nas quais as dúvidas são esclarecidas e é realizado o monitoramento do progresso do trabalho [26] [34].

Um dos métodos ágeis que tem ganhado popularidade devido sua simplicidade, produtividade e gerenciamento leve é o *Scrum* [39]. O *Scrum* possui a ideia central de que muitos processos não podem ser previstos durante o desenvolvimento do sistema e por isto ele adota uma abordagem flexível [44]. O principal produto do *Scrum* é o *Product Backlog* (PB), que é uma lista de todos os requisitos priorizados. Esses requisitos são selecionados e planejados em uma *Sprint Planning* e formam o *Sprint Backlog* (SB). Com o SB se inicia a *Sprint* que tem duração fixa. Ao decorrer dela, ocorrem as reuniões diárias para alinhamento do time. No final da *Sprint* ocorrem as reuniões de *Review* nas quais é apresentado o resultado da *Sprint*. Logo em seguida é feita a *Retrospective* para identificar os pontos positivos e os pontos a serem melhorados [45].

Existem os seguintes papéis no *Scrum* – os *Stakeholders* que são os responsáveis pela ideia do que construir e seu porquê; *Product Owner* (PO) responsável pela criação do PB; *Scrum Master* é o facilitador, garante que a entrega do produto será feita com êxito e o *Scrum Team* são os desenvolvedores do produto [45].

2.2 Engenharia de Requisitos

A Engenharia de Requisitos (ER) é um processo para estabelecer os serviços requeridos pelo cliente a respeito de um sistema. Por meio dela, são identificadas as necessidades e as restrições de desenvolvimento [26]. A ER está preocupada com a identificação, modelagem, comunicação e documentação dos requisitos de acordo com contexto que o sistema será utilizado [33].

A comunidade de desenvolvimento ágil sabe pouco sobre os papeis, processos e práticas da ER tradicional, assim ela adota uma abordagem mais flexível e dinâmica de trabalho [17]. O principal objetivo do processo de ER tradicional é criar uma documentação para compartilhar o conhecimento, enquanto no desenvolvimento ágil o foco está em uma comunicação face a face entre o cliente e os times ágeis para atingir o mesmo objetivo [26].

Apesar da importância da ER no sucesso do desenvolvimento do software e na minimização dos riscos de projeto, essa atividade é vista, algumas vezes, nos métodos ágeis como burocrática, podendo tornar o processo menos ágil. A principal justificativa ocorre devido ao fato de que os requisitos mudam tão rapidamente que um documento de requisitos fica desatualizado tão logo seja redigido, desperdiçando todo esforço empreendido na documentação [19].

As atividades de ER (elicitação, documentação, validação e gerenciamento) não são atividades claramente definidas na ER ágil [36]. O estudo de Lucia e Qusef [26] realizou uma comparação das atividades da ER tradicional implementadas no método ágil Scrum, resumidas na Tabela 1. As atividades da ER tradicional são de responsabilidade do Analista de Requisitos enquanto no *Scrum*, os requisitos são definidos e priorizados pelo PO. Os métodos apresentam diferenças na execução de cada atividade, conforme mostrado na Tabela 1. Uma destas diferenças é o fato dos métodos ágeis trabalharem com requisitos descritos e gerenciados por meio de histórias de usuário, descritas na próxima seção.

2.3 Histórias de Usuário

A história de usuário é definida por uma ou duas frases do cotidiano ou do negócio do usuário final do sistema, a fim de capturar a essência do que deve ser desenvolvido [8]. Os métodos ágeis utilizam as histórias de usuário para representar a necessidade do cliente. As histórias devem ser simples, objetivas e conter o benefício que será entregue. As histórias de usuário têm o objetivo de ancorar uma discussão futura. Os requisitos são discutidos de forma detalhada antes e/ou durante a implementação. Entretanto, não necessariamente e, às vezes, raramente esses levantamentos são documentados. A comunicação face a face, em alguns projetos, substitui a necessidade de requisitos definidos e documentados formalmente, salvo se houver uma obrigação para tal [34].

Tabela 1. Reimplementação ER no *Scrum* (Adaptado de [26]**)**

Atividades da ER	ER Tradicional	*Scrum*
Elicitação de Requisitos	Analista de Requisitos (AR) levanta os requisitos como cliente.	• PO cria o PB; • Alguns *stakeholders* podem participar da criação do PB.
Análise de Requisitos	AR retira as dúvidas com o cliente e analisa a viabilidade dos requisitos.	• Reunião para o refinamento do *Backlog*; • PO prioriza o PB; • PO analisa a viabilidade dos requisitos.
Documen-tação de Requisitos	O AR documenta os requisitos através de casos de uso e diagramas.	• Comunicação face a face;
Validação dos Requisitos	O AR solicita que o cliente valide o entendimento dos requisitos.	• Reunião de *Review*.
Gerência de Requisitos	Acompanhar desenvolvimento dos requisitos. Alterações nos requisitos em função de controle	• Reunião de Planejamento da *Sprint*; • Acompanhar itens do PB; • Mudanças de requisitos são

	rígido de mudanças.	incluídas ou excluídas do PB.

2.4 Engenharia de Requisitos Ágil

O estudo realizado por Rubin e Rubin [35] identificou que, para se trabalhar com documentação nos métodos ágeis, é necessária a colaboração de toda a equipe e, além disso, é preciso que todos os envolvidos tenham um conhecimento e uma linguagem comum para a comunicação efetiva. Já o estudo de Ramesh, Cao, e Baskerville [34] identificou seis práticas relacionadas à ER ágil que são: comunicação face a face ao invés de documentar, ER interativa, gerenciar a mudança dos requisitos através de um planejamento constante, frequente priorização dos requisitos, prototipação, reuniões de revisão/testes.

Em termos de produtos de trabalho, Schön, Thomaschewski, e Escalona [36] identificaram que os principais artefatos utilizados nos métodos ágeis são: histórias de usuário, protótipos, casos de uso e cartões de histórias. Outros autores, [14], [30] e [36], citam que os requisitos podem estar documentados nos casos de testes. Apesar desta variedade de produtos de trabalho, quando há prejuízo da documentação em projetos que usam métodos ágeis, conhecimentos importantes podem ser perdidos durante e após o desenvolvimento do sistema [35]. Neste contexto, a revisão sistemática apresentada por Jaqueira e outros [19] identificou os desafios mais citados relacionados a requisitos em métodos ágeis, sendo os mais relevantes: disponibilidade do cliente para fornecer feedback constante; rastreabilidade das frequentes mudanças; requisitos não funcionais negligenciados; consenso na negociação ou priorização dos requisitos, principalmente quando há mais de um cliente e falta de documentação adequada.

O trabalho realizado pelos autores Inayat e outros [17] teve o objetivo de mostrar como os problemas da ER tradicional são resolvidos com o desenvolvimento ágil. A metodologia utilizada pelos autores foi realizar uma revisão sistemática de práticas e desafios da ER em métodos ágeis. A conclusão obtida pelos autores foi que as práticas da ER ágil ainda estão imaturas e necessitam de mais estudos de caso. Eles também concluíram que uma junção das práticas da ER tradicional e ágil pode trazer resultados muito positivos. Por fim, os autores apresentaram uma lista de desafios da área.

Os trabalhos relacionados tiveram enfoques diferentes do presente estudo. Alguns apresentaram artefatos da ER usados em métodos ágeis [36], alguns identificaram práticas, porém por meio de estudo em empresas [34] e outros focaram em desafios da adoção destas práticas [17]. Além de abranger todas as atividades da ER, este trabalho identificou por meio de uma revisão sistemática não somente práticas e técnicas de todas as atividades da ER, mas também identificou benefícios alcançados pelos métodos ágeis e ainda caracterizou os trabalhos de pesquisa da área.

3. METODOLOGIA

A metodologia deste trabalho consiste em uma revisão sistemática, a fim de identificar as práticas de ER utilizadas nos requisitos ágeis.

Uma revisão sistemática se inicia com a formulação de perguntas a serem respondidas. O objetivo desta revisão é responder às perguntas: P1 – Quais práticas de levantamento e detalhamento de requisitos as organizações têm seguido? P2 – Quais benefícios as empresas têm alcançado com o uso de tais práticas e quais são os desafios a serem vencidos? P3 – Como pode ser caracterizada a pesquisa em Engenharia de Requisitos aplicada a métodos ágeis?

Em seguida, as buscas foram realizadas nas bases ACM Digital Library, IEEEXplore Digital Library, Science Direct. A string de busca utilizada foi "((Requirements Engineering) OR (Software requirements) OR (Agile requirements) OR (Requirements documentation)) AND ((Agile methods) OR (Agile methodologies))". As buscas retornaram 277 artigos, excluindo os duplicados e ao final foram selecionados 34.

Para determinar os trabalhos incluídos e excluídos neste estudo, foram definidos critérios baseados nos artigos [17] e [19]. Como critérios de inclusão, foram definidos: os trabalhos devem apresentar usos da ER em métodos ágeis; soluções ou adaptações da ER tradicional para métodos ágeis ou tratar de problemas de ER em métodos ágeis. Como critérios de exclusão, foram excluídas publicações diferentes de artigos científicos de periódicos e congressos, como resumos, tutoriais, resenhas e tutoriais; e ainda estudos que não tratem da ER em métodos ágeis. O ano de publicação do artigo não foi utilizado nem como critério de inclusão nem como critério de exclusão.

O processo de revisão sistemática se iniciou pela análise dos títulos dos trabalhos retornados pela pesquisa a fim de descartar os estudos que não estavam diretamente relacionados com os objetivos da revisão. Neste ponto, foram analisados 277 artigos e descartaram-se 184. Em uma segunda fase, foram lidos os resumos dos 93 artigos restantes para garantir que os critérios de inclusão estavam sendo satisfeitos. Por último, foi feita a leitura completa dos 49 trabalhos que restaram, novamente com a aplicação dos critérios de inclusão e exclusão descritos anteriormente. Ao final, restaram 34 trabalhos para serem analisados na revisão sistemática. Os trabalhos que atenderam aos critérios de inclusão da revisão foram publicados entre os anos de 2003 e 2016.

4. RESULTADOS

Os resultados da revisão sistemática foram obtidos a partir da categorização dos 34 estudos lidos por completo. Esta classificação foi feita buscando categorizar as metodologias de pesquisa, as técnicas de elicitação e documentação de requisitos, os benefícios e as lições aprendidas, os desafios e problemas e, por fim, as ferramentas (software) utilizadas ou apresentadas nestes estudos, cumprindo assim os objetivos da pesquisa.

A Tabela 2 mostra as principais metodologias de pesquisa utilizadas nos trabalhos. Analisando as metodologias listadas, percebe-se que os estudos de caso foram as pesquisas mais frequentes, conforme outras revisões sistemáticas realizadas [15], [17] e [36]. Assim, em resposta à pergunta P3, as pesquisas em ER aplicadas em métodos ágeis são caracterizadas, principalmente, pelo uso de estudos de caso e entrevistas.

Em consideração à pergunta P1, mais especificamente em relação às técnicas de ER utilizadas, a Tabela 3 apresenta as técnicas de elicitação mais citadas e utilizadas nos estudos levantados. A lista de técnicas foi baseada no trabalho dos autores Carrizo, Dieste, e Juristo [3] e usada como base para fazer a análise dos trabalhos lidos.

Tabela 2. Distribuição dos Métodos de Pesquisa

Metodologia de Pesquisa	Qtde.	Percentual de artigos
Estudo de Caso	18	52,9%

Entrevista	12	35,3%
Questionário	10	29,4%
Estudo Exploratório	10	29,4%
Construção de Modelos	5	14,7%
Revisão Sistemática	5	14,7%

A Tabela 4 apresenta uma análise dos trabalhos em função das técnicas e práticas das outras fases da ER (Análise, Documentação e Validação). Foi também identificada a adoção de práticas gerais em conjunto com as de ER. A característica de maior de frequência foi a de gerência de requisitos e mudanças, a qual esteve presente em quase todos os artigos analisado. Isto se justifica pelo fato dos requisitos estarem em constante mudança e é preciso ter controle para que informações não sejam construídas erroneamente. E para gerenciar com sucesso, o envolvimento do cliente é essencial, uma prática frequente identificada nos estudos.

Tabela 3. Técnicas de Elicitação

Técnica de Elicitação	Qtde.	Referências dos trabalhos que usam a técnica de elicitação
Histórias de usuário	25	[2][7][8][12][13][14][15][16][17][18][20][21][22][23][24][25][26][27][28][31][36][38][40][43][44]
Entrevistas	12	[1][8][13][15][18][21][26][27][30][31][36][37]
Cenários	10	[7][14][22][26][28][27][30][36][38][43]
Workshop	9	[8][13][15][17][18][21][26][28][30]
Observação Participante	7	[8][14][15][22][26][27][41]
Brainstorming	5	[16][26][27][36][37]
Questionário	4	[8][13][18][22]
Grupo Focal	3	[1][16][26]
JAD	3	[6][26][30]

Com relação a documentação de requisitos, verifica-se que as histórias de usuários estão à frente dos casos de uso. Os modelos visuais (diagramas de requisitos e os de negócio) são frequentemente citados/utilizados, isto se deve ao fato da facilidade de entendimento do artefato.

Os protótipos também ocupam uma posição de destaque, já que eles, conforme os autores Käpyaho e Kauppinen [20], ajudam a enfrentar alguns desafios da engenharia de requisitos ágil. Diversos estudos apresentam o uso de uma ferramenta de apoio às atividades de ER ([8], [10], [12], [16], [17], [21], [31] e [40]). Foram encontradas referências a 21 ferramentas diferentes, sendo poucas repetidas em mais de um artigo. As ferramentas mais frequentemente citadas foram a Wiki e o software de gestão de problemas Jira, e a Version One. Percebe-se uma diversidade de ferramentas para gerenciar os requisitos ágeis.

Tabela 4. Práticas Ágeis e Técnicas da ER

Geral	Técnica/ Prática	Qtde.	Referências dos artigos que utilizam a prática ou técnica
Práticas Ágeis	Envolvimento do Cliente	26	[1][2][4][7][8] [9][13][14][15][16][17][18][20][21][22][23][24][25][26][27][30][31][36][38][37][40]
	Compartilhamento de Conhecimento	19	[1][6][7][9][13][16][17][18][20][22][23][26][30][28][36][40][41][43][44]
	Comunicação face-face	15	[1][6][12][13][14][15][16][17][18][21][25][28][33][36][40]
	Conhecimento Tácito	10	[1][7][16][17][22][25][30][36][37][43]
Análise de Requisitos	Priorização/ Dependên-cias de Requisitos	21	[2][7][8][9][13][14][15][16][17][18][21][22][23][26][31][30][36][37][38][43][44]
	Valor para o Negócio	9	[7][13] [14][15][22][24] [25][26][44]
Documentação de Requisitos	Modelos/Diagramas de Requisitos e Negocio	16	[1][6][9][10][14][16][17][18][21][23][26][27][30][36][41][43]
	Protótipos	11	[1][6][13][17][26][27][23][24][28][36][37]
	Casos de Uso/ Especificações	10	[12][14][16][20][23][26][27][36][38][43]
	Casos de Teste	3	[14][30][36]
	Comentário no código fonte	2	[6][40]
Validação de Requisitos	Gerência de Requisitos /Mudanças	31	[1][2][4][7][8][9][10][12][13][14][15][16][17][18][20][21][23][24][25][27][28][30][31][33][36][38][37][40][41][42][44]
	Validação de Requisitos	14	[4][7][10][16][17][18][21][22][23][24][26][30][31][38]
	Testes de Aceitação	13	[7][8][12][14][15][16][17][21][22][25][30][36][41]

Em relação à Pergunta P2, a Tabela 5 descreve os benefícios mais relevantes reportados nos trabalhos analisados. O benefício mais frequente foi a melhora da comunicação e o envolvimento do

time, isto se deve ao fato do conhecimento ser compartilhado continuamente.

O segundo benefício foi o feedback rápido, o qual é de extrema importância para validar se o entendimento dos requisitos está correto e se o sistema irá atender as necessidades do usuário. Outro benefício que deve ser destacado é a melhora no entendimento os requisitos. Apesar de vários benefícios reportados, há trabalhos que observaram em suas lições aprendidas criticando o fato de que a documentação em projetos ágeis nem sempre é suficiente e que existe débito de documentação de requisitos, por exemplo esses trabalhos: [20], [21], [28], [31], [38] e [40].

Tabela 5. Benefícios e Lições aprendidas

Benefício/Lição Aprendida	Qtde.	Referências
Melhoria da comunicação/ envolvimento das partes interessadas	16	[1][4][6][12][13] [14][16][21][22] [24][25][27][28] [30][33][37]
Feedback rápido	14	[7][9][12][13][14] [16][17][21][22] [23][24][25][26] [36]
Redução do volume da documentação	10	[2][6][12][16][17][18][23][28][30][33]
Redução da ambiguidade	9	[9][10][13][20][21] [22][23][33][37]
Melhoria da visão das funcionalidades por meio de prototipação	9	[6][13][15][17][21] [28][30][36][38]
Melhoria do entendimento dos requisitos	9	[1][14][15][21][22] [23][27][37][40]
Aumento de inovação e criatividade	6	[13][21][22][27] [28][37]
Aumento do débito de requisitos	6	[20][24][28][31] [40][41]
Entrega rápida e constante	5	[4][15][25][28] [30]
Aumento da responsividade a mudanças	5	[1][15][16][21][24]
Aumento da facilidade de se decompor requisitos complexos	5	[14][24][25][37] [44]

Na Tabela 6 são descritos os principais problemas e desafios enfrentados e influenciados pela ER ágil. O desafio mais recorrente é a comunicação, pois é preciso evitar o "mal entendimento" de uma necessidade complexa. Apesar disso, a melhora da comunicação é um benefício, citado na Tabela 5. Isso, apesar de parecer contraditório, se justifica pela dificuldade que é a comunicação entre as pessoas com diferentes níveis de conhecimento, podendo levar a débitos de documentação. Uma vez que os requisitos são disponibilizados através do conhecimento, construído pelo time e não por uma documentação [40].

Outro desafio recorrente nos estudos foi a negligência dos requisitos não funcionais, os quais, muitas vezes são ignorados e se tornam dependentes do conhecimento básico do time. Este conhecimento básico também um é desafio, que ocupa a quinta posição nos estudos lidos.

O tipo do artefato a ser utilizado para divulgar os requisitos entre o time também é um desafio, frente as várias formas existentes (prototipação, casos de uso, histórias de usuários, diagramas, etc.). Outros desafios citados foram estimativas imprecisas, dificuldades de priorização, a disponibilidade do cliente, que também acontecem na ER tradicional.

Tabela 6. Desafios e Problemas

Desafios e Problemas	Qtde.	Referências
Falhas de comunicação	16	[7][12][14][16][17] [18][21][20][23][24][27][31][36][38][40] [41]
Negligência dos requisitos não funcionais	13	[7][13][14][15][17] [20][21][28][30][31] [33] [36][38]
Pouca documentação	10	[14][15][17][20][21] [28][31][36][38][40]
Histórias de usuário simplórias	7	[14][15][24][28][31] [33][40]
Possuir Conhecimento básico	7	[10][23][31][33][38] [42][43]
Tipo do artefato utilizado para comunicação	6	[7][15][16][30][36] [41]
Estimativas imprecisas	6	[15][20][24][28][38] [31]
Dificuldades de Priorização	4	[14][15][21][31]
Disponibilidade do cliente	4	[15][17][20][31]

5. DISCUSSÕES E CONCLUSÕES

Este trabalho teve como objetivo identificar um panorama da utilização de práticas e técnicas de Engenharia de Requisitos em estudos que utilizam métodos ágeis. Tal identificação foi realizada por meio de uma revisão sistemática da literatura. Foi possível identificar as técnicas de elicitação de requisitos mais utilizadas, como os processos de análise, documentação e detalhamento, verificação e validação são realizados. Por fim, foram identificados benefícios e desafios relacionados à Engenharia de Requisitos ágil.

O Manifesto Ágil prevê que deve ser estabelecido um contato frequente com o cliente. Neste trabalho, percebeu-se que as técnicas de elicitação mais frequentemente citadas no contexto ágil foram as entrevistas e cenários. De acordo com Carrizo, Dieste, e Juristo [3], tais técnicas são mais adequadas quando aplicadas a um informante individualmente (neste caso, o PO), quando há a necessidade de consenso entre os envolvidos, o

interesse do informante é alto, ele possui disponibilidade de tempo e está localizado próximo aos entrevistadores. Todas estas características são frequentemente observadas no uso de métodos ágeis.

Quanto às práticas gerais identificadas, sobressaíram-se práticas dos métodos ágeis que possuem forte influência na ER, tais como, envolvimento do cliente, comunicação face a face e conhecimento tácito. Observa-se ainda que quase metade dos estudos identificados aborda formas de documentação mais detalhadas de requisitos, como modelos e diagramas ou ainda casos de uso. Tal fato pode ser indicativo de que as empresas utilizam formas consideradas mais tradicionais na documentação de requisitos. Poucos relatos foram identificados quanto à documentação de requisitos em casos de testes.

Em métodos ágeis, em função do envolvimento constante do cliente, atividades de validação são frequentes. Este estudo identificou a referência à validação de requisitos e testes de aceitação em quase metade dos estudos encontrados. A Gerência de Requisitos é citada em praticamente todos os estudos.

Benefícios reportados pelo uso de métodos ágeis relacionados às atividades de ER envolvem melhoria da comunicação e compreensão das necessidades, redução da ambiguidade e capacidade de responder a mudanças. Foram identificadas ainda lições aprendidas reportados nos trabalhos analisados, tais como, o fato de se utilizar diferentes técnicas de elicitação auxilia no levantamento dos requisitos ([8], [22], [30] e [36]); a adoção de práticas da ER tradicional e com práticas ágeis traz resultados positivos ([7], [14], [15] e [17]); assim como faz surgir um perfil de profissional com características híbridas entre os dois modelos ([7] e [44]).

Os desafios mostram possíveis caminhos a serem trabalhados como o débito de documentação, dificuldades de se trabalhar com histórias de usuário simplórias e falhas de comunicação. Tais resultados estão em conformidade com os encontrados por Chow e Cao [5] que apresenta fatores críticos de sucesso em projetos ágeis de software, tais como falta de capacidades e habilidades da equipe, ausência de trabalho em equipe; requisitos mal definidos e ausência do cliente. Acrescentando a esta lista, o presente estudo identificou ainda que é outros fatores que dificultam o uso das práticas da ER ágil são a grande variedade de técnicas e práticas possíveis a serem usadas na ER ágil ([14], [16], [25] e [36]) e a importância de se manter a rastreabilidade entre os artefatos criados ([12], [23] e [43]).

Apesar desta pesquisa ter procurado abranger importantes bases científicas, uma limitação do trabalho está no fato de a revisão sistemática ter sido realizada em três bases científicas. Assim, estender a pesquisa para outras bases é uma forma de ampliar a caracterização realizada. Não foi realizada tampouco uma análise da aplicação das práticas de ER ágil no contexto dos diversos métodos existentes. Sendo assim, mapear as práticas identificadas neste trabalho para as diferentes metodologias ágeis é um trabalho ainda a ser feito. Além disso, a identificação das práticas de ER foi realizada com base exclusivamente na revisão da literatura. Uma análise de como as empresas de fato utilizam as práticas e técnicas de ER em projetos com métodos ágeis e quais resultados têm sido alcançados é de grande valia e, por isso, considerada um trabalho futuro. Desta forma, será possível ainda confrontar a literatura publicada com as práticas das empresas desenvolvedoras de software.

6. REFERÊNCIAS

[1] Abdullah, N.N.B. et al. 2011. Communication patterns of agile requirements engineering. *Proceedings of the 1st Workshop on Agile Requirements Engineering*. (jul. 2011), 1–4.

[2] Baruah, N. 2015. Requirement management in agile software environment. *Proceedings of the 2015 International Conference on Soft Computing and Software Engineering*. 62, (dez. 2015), 81–83.

[3] Carrizo, D. et al. 2008. Study of elicitation techniques adequacy. *Proceedings of the 11th Workshop on Requirements Engineering*. 11, (jan. 2008), 104–114.

[4] Chen, J.Q. et al. 2007. Light-Weight Development Method: a Case Study. *International Conference on Service Systems and Service Management*. (jun. 2007), 1–6.

[5] Chow, T. e Cao, D. 2008. A survey study of critical success factors in agile software projects. *Journal of Systems and Software*. 81, (ago. 2008), 961–971.

[6] Clear, T. 2003. Documentation and agile methods. *SIGCSE Bulletin*. 35, 2 (jun. 2003), 12–13.

[7] Daneva, M. et al. 2013. Agile requirements prioritization in large-scale outsourced system projects: An empirical study. *Journal of Systems and Software*. 86, 5 (maio 2013), 1333–1353.

[8] Dimitrijević, S. et al. 2015. A comparative study of software tools for user story management. *Information and Software Technology*. 57, (jan. 2015), 352–368.

[9] Dragicevic, S. et al. 2014. Use of method for elicitation, documentation, and validation of software user requirements (MEDoV) in agile software development projects. *Proceedings - 6th International Conference on Computational Intelligence, Communication Systems and Networks*. (maio 2014), 65–70.

[10] Ernst, N.A. et al. 2014. Agile requirements engineering via paraconsistent reasoning. *Information Systems*. 43, (jul. 2014), 100–116.

[11] Gandomani, T.J. et al. 2013. Obstacles in moving to agile software development methods; At a Glance. *Journal of Computer Science*. 9, 5 (jan. 2013), 620–625.

[12] Gayer, S. et al. 2016. Lightweight Traceability for the Agile Architect. *Computer*. 49, 5 (maio 2016), 64–71.

[13] Ghanbari, H. et al. 2015. Utilizing online serious games to facilitate distributed requirements elicitation. *Journal of Systems and Software*. 109, (nov. 2015), 32–49.

[14] Haugset, B. e Stålhane, T. 2012. Automated acceptance testing as an agile requirements engineering practice. *Hawaii International Conference on System Sciences*. 45, (jan. 2012), 5289–5298.

[15] Heikkila, V.T. et al. 2015. A Mapping Study on Requirements Engineering in Agile Software Development. *Euromicro Conference on Software Engineering and Advanced Applications*. 41, (ago. 2015), 199–207.

[16] Hoda, R. et al. 2010. How much is just enough? Some Documentation Patterns on Agile Projects. *Proceedings of the 15th European Conference on Pattern Languages of Programs*. 15, (jul. 2010), 1–13.

[17] Inayat, I. et al. 2015. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*. 51, (out. 2015), 915–929.

[18] Inayat, I. et al. 2012. Socio-technical aspects of requirements-driven collaboration (RDC) in agile software development methods. *IEEE Conference on Open Systems*. (out. 2012), 1–6.

[19] Jaqueira, A. et al. 2012. Desafios de Requisitos em Métodos Ágeis: Uma Revisão Sistemática. *Brazilian Workshop on Agile Methods*. 3, (jan. 2012), 49–60.

[20] Kajko-Mattsson, M. 2008. Problems in agile trenches. *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*. (out. 2008), 111–119.

[21] Käpyaho, M. e Kauppinen, M. 2015. Agile requirements engineering with prototyping: A case study. *International Requirements Engineering Conference*. 23, (ago. 2015), 334–343.

[22] Kelly, S. 2010. Towards an evolutionary framework for agile requirements elicitation. *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*. (jun. 2010), 349–352.

[23] Kumar, M. et al. 2010. Towards knowledge assisted agile requirements evolution. *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*. (maio 2010), 16–20.

[24] Liskin, O. et al. 2014. Understanding the role of requirements artifacts in kanban. *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering*. (jun. 2014), 56–63.

[25] Liu, L. e Lu, Y. 2012. Application of agile method in the enterprise website backstage management system: Practices for extreme programming. *International Conference on Consumer Electronics, Communications and Networks (CECNet)*. 2, (abr. 2012), 2412–2415.

[26] Lucia, A. De e Qusef, A. 2010. Requirements Engineering in Agile Software Development. *Journal Of Emerging Technologies In Web Intelligence*. 2, 3 (ago. 2010), 212–221.

[27] Mahmud, I. e Veneziano, V. 2011. Mind-mapping: An effective technique to facilitate requirements engineering in agile software development. *International Conference on Computer and Information Technology*. 14, (dez. 2011), 157–162.

[28] Maiden, N. e Jones, S. 2010. Agile Requirements Can We Have Our Cake and Eat It Too? *IEEE Software*. 27, 3 (maio 2010), 87–88.

[29] Manifesto for Agile Software Development: 2001. *http://agilemanifesto.org/*.

[30] Mead, N.R. et al. 2008. Incorporating Security Requirements Engineering into the Dynamic Systems Development Method. *32th Annual IEEE International Computer Software and Applications Conference*. 32, (ago. 2008), 949–954.

[31] Mendes, T.S. et al. 2016. Impacts of Agile Requirements Documentation Debt on Software Projects : A Retrospective Study. *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. 31, (abr. 2016), 1290–1295.

[32] Mishra, D. e Mishra, A. 2011. Pattern detection for conceptual schema recovery in data-intensive systems. *Journal of Software: Evolution and Process*. 23, (jan. 2011), 549–564.

[33] Paetsch, F. et al. 2003. Requirements engineering and agile software development. *Proceedings. 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. 12, (jun. 2003), 1–6.

[34] Ramesh, B. et al. 2010. Agile requirements engineering practices and challenges: an empirical study. *IEEE Software*. 25, 1 (fev. 2010), 449–480.

[35] Rubin, E. e Rubin, H. 2011. Supporting agile software development through active documentation. *Requirements Engineering*. 16, 2 (jun. 2011), 117–132.

[36] Schön, E.-M. et al. 2017. Agile Requirements Engineering: A Systematic Literature Review. *Computer Standards & Interfaces*. 49, (jan. 2017), 79–91.

[37] Sillitti, A. et al. 2005. Managing Uncertainty in Requirements: a Survey in Documentation-driven and Agile Companies. *IEEE International Software Metrics Symposium*. 11, (set. 2005), 1–10.

[38] Soares, H.F. et al. 2015. Investigating the Link between User Stories and Documentation Debt on Software Projects. *International Conference on Information Technology - New Generations*. 12, (abr. 2015), 385–390.

[39] Stankovic, D. et al. 2013. A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*. 86, 6 (jun. 2013), 1663–1678.

[40] Stettina, C.J. e Heijstek, W. 2011. Necessary and Neglected? An Empirical Study of Internal Documentation in Agile Sotware Development Teams. *Proceedings of the 29th ACM international conference on Design of communication*. 29, (out. 2011), 159–166.

[41] Stettina, C.J. e Kroon, E. 2013. Is there an agile handover? An empirical study of documentation and project handover practices across agile software teams. *International Conference on Engineering, Technology and Innovation, ICE 2013 and IEEE International Technology Management Conference*. (jun. 2013), 1–12.

[42] Tarhan, A. e Yilmaz, S.G. 2014. Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process. *Information and Software Technology*. 56, 5 (maio 2014), 477–494.

[43] Trkman, M. et al. 2016. Using business process models to better understand the dependencies among user stories. *Information and Software Technology*. 71, (mar. 2016), 58–76.

[44] Vlaanderen, K. et al. 2011. The agile requirements refinery: Applying SCRUM principles to software product management. *Information and Software Technology*. 53, 1 (jan. 2011), 58–70.

[45] Wan, J. et al. 2013. Case Study on Critical Success Factors of Running Scrum. *Journal of Software Engineering and Applications*. 6, (fev. 2013), 59–64.