

---

## **Uma comparação entre achados da indústria e da literatura em relação ao uso de requisitos de qualidade em projetos ágeis**

---

Nikollas Filgueiras da Silva  
Centro de Informática,  
Universidade Federal de Pernambuco  
Recife, Pernambuco, Brasil  
Email: nfs2@cin.ufpe.br

Carla Taciana Lima Lourenço Silva Schuenemann  
Centro de Informática,  
Universidade Federal de Pernambuco  
Recife, Pernambuco, Brasil  
Email: ctlls@cin.ufpe.br

**Resumo:** Com um mercado bastante competitivo, em meados de 2001 surge o Manifesto Ágil, com o objetivo de focar nas entregas contínuas em vez da formalização exagerada nas documentações. Os métodos ágeis também visam promover a união entre profissionais do negócio e desenvolvedores, além de tornar mais fácil a incorporação de mudanças nos requisitos. Para alcançar um melhor alinhamento entre a Engenharia de Requisitos, mais especificamente a Engenharia de Requisitos de Qualidade, e as Metodologias Ágeis, é necessário que algumas práticas que são vivenciadas atualmente sejam revistas, pois as mesmas acabam contribuindo com a negligência desse tipo de requisitos nos projetos ágeis de software. Nesse cenário, foi realizado um estudo cujo objetivo foi identificar os desafios na engenharia de requisitos de qualidade em projetos ágeis de larga escala e distribuídos que foram pesquisados até o momento, as práticas ágeis que contribuíram para o surgimento desses desafios e as soluções propostas. Com base nos resultados desse estudo, o presente trabalho visa averiguar como os desafios e práticas que fazem parte da engenharia de requisitos de qualidade em projetos ágeis estão sendo enfrentados na indústria brasileira de desenvolvimento de software. Portanto, este trabalho apresenta os resultados de uma pesquisa realizada por meio de um survey que foi respondido por diversos profissionais que trabalham em projetos de software e oriundos de várias partes do país. Através de suas respostas foi possível identificar quais os desafios enfrentados pelo mercado brasileiro em relação ao uso de Requisitos de Qualidade juntamente com as Metodologias Ágeis. Posteriormente, os dados obtidos do questionário foram cruzados com as respostas das perguntas de pesquisa presentes na revisão sistemática de literatura produzida por outros autores. Como resultado, o trabalho apresenta a identificação de linhas de pesquisas que poderão beneficiar o mercado no futuro.

**Palavras chave:** Engenharia de Requisitos, Requisitos de Qualidade, Metodologias Ágeis, Desafios.

---

## 1 Introdução

Vivemos em um cenário profissional bastante voraz e competitivo e, por causa disso, é extremamente necessário que as empresas estejam sempre atualizadas para entregar projetos eficazes e de forma rápida. É necessário que a empresa bem-sucedida adote novas formas de gerenciamento, inove, seja ágil para atingir o planejamento do empreendimento (Opservices, 2015).

Diante disso, em 2001 ocorreu a publicação do Manifesto Ágil, que foi um marco para o Desenvolvimento Ágil de Software. Um grupo de 17 especialistas propôs práticas para melhorar o desenvolvimento de software, definindo valores e princípios para guiar os métodos ágeis. Eles consideraram que as abordagens tradicionais eram pesadas e o processo muito burocrático. O Manifesto Ágil propõe a adoção de processos flexíveis e adaptáveis a aceitar as mudanças como parte inseparável de seu processo de desenvolvimento (VERSIONONE, 2016 apud MEDEIROS, 2017).

As metodologias ágeis de desenvolvimento de software, ou Agile, surgiram e ganharam muita força nos últimos anos devido à necessidade do mercado em atender às demandas dos clientes e seus projetos de maneira mais dinâmica, flexível e com maior produtividade. Em um cenário de transformações constantes, a escolha pelo paradigma de desenvolvimento mais adequado a um determinado contexto é, sem sombra de dúvidas, um fator crucial para o sucesso do projeto. As primeiras técnicas formais voltadas à construção de sistemas possuíam um enfoque direcionado à elaboração de soluções com uma estrutura mais rígida e, portanto, eram menos suscetíveis a mudanças. A necessidade de uma rápida adaptação diante de situações imprevistas é um dos aspectos que caracterizam o mundo corporativo atual; Procurando atender a este tipo de demanda, surgiram práticas mais flexíveis para a criação de aplicações, com diversas dessas sendo conhecidas como “metodologias ágeis” (GROFFE, 2018). Em uma entrevista feita pela VersionOne em 2015, com cerca de 4000 pessoas, mostrou-se que 45% dos entrevistados utilizavam metodologias ágeis em seus projetos (VERSIONONE, 2015 apud MEDEIROS, 2017).

No processo de desenvolvimento de software, uma das etapas de grande importância é a Engenharia de Requisitos. Requisitos têm um papel central no processo de software, sendo considerado um fator determinante para o sucesso ou fracasso de um projeto de software (SOMMERVILLE, 2011). O processo de Engenharia de Requisitos envolve criatividade, interação de diferentes pessoas, conhecimento e experiência para transformar informações diversas (sobre a organização, sobre leis, sobre o sistema a ser construído, etc.) em

documentos e modelos que direcionam o desenvolvimento do software (ARRUDA; SOARES; VIEIRA, 2013).

Uma das diferenças entre um modelo Tradicional e um Ágil é a capacidade de responder a mudanças de forma evolutiva e de imediato. Com isso é preciso ficar atento às mudanças de requisitos. Para haver uma abordagem eficiente entre a Engenharia de Requisitos aplicada em metodologias ágeis, faz-se necessário que os procedimentos utilizados em Engenharia de Requisitos sejam repensados, pelo fato de que tais metodologias podem abdicar de certos documentos e artefatos utilizados em processos tradicionais (ALVES, 2009).

Requisitos de qualidade (Quality Requirements ou QRs) são os requisitos que descrevem as qualidades do sistema (por exemplo, requisitos de desempenho, requisitos de manutenção). Na literatura da Engenharia de Requisitos (ER), há um consenso de que o sucesso ou o fracasso de um sistema não é decidido apenas pela implementação dos requisitos funcionais corretos, mas também pela correta implementação dos QRs. Por exemplo, se o tempo de resposta do sistema não atender às expectativas do cliente, não podemos dizer que o sistema oferece qualidade (ALSAQAF; DANEVA; WIERINGA, 2017).

Nesse cenário, a revisão sistemática de literatura intitulada de “Quality Requirements in Large-Scale Distributed Agile Projects – A Systematic Literature Review” foi realizada (ALSAQAF; DANEVA; WIERINGA, 2017). Ela foi motivada pela percepção dos autores de que os métodos de desenvolvimento ágeis, apesar de populares, dificilmente atentam para os requisitos de qualidade, o que poderia prejudicar os lucros da entrega rápida através da introdução de altos esforços de retrabalho posteriormente. Segundo os autores, já existem várias publicações que relataram a atenção insuficiente aos requisitos de qualidade em métodos ágeis. Mesmo conhecendo as práticas ágeis de ER e seu impacto nos requisitos de qualidade, as empresas usam cada vez mais os métodos ágeis em ambientes distribuídos de larga escala, tendo como resultado negativo o fato de ignorar cada vez mais os requisitos de qualidade. Com isso, a pesquisa teve como objetivo identificar os desafios na Engenharia Ágil de Requisitos de Qualidade em projetos distribuídos de grande escala, relatados em artigos científicos, e as soluções propostas.

Com base nos resultados da revisão sistemática da literatura (RSL) realizada por Alsaqaf, Daneva & Wieringa (2017), surgiu a motivação de averiguar quais os desafios que fazem parte da Engenharia de Requisitos de Qualidade estão sendo enfrentados juntamente com o uso de metodologias ágeis na indústria brasileira de desenvolvimento de software. Portanto, este trabalho de conclusão de curso tem como objetivo fazer uma comparação entre achados da indústria e da literatura em relação ao uso de requisitos de qualidade em projetos ágeis.

Com este propósito, é necessário realizar um survey na indústria brasileira de desenvolvimento de software para elencar os desafios enfrentados pelo uso da engenharia de requisitos em projetos ágeis de software. Além disso, o trabalho vai comparar os resultados do survey com as respostas de duas das três perguntas de pesquisa presentes na RSL dos autores Alsaqaf, Daneva & Wieringa (2017). O motivo de não abordar a última pergunta de pesquisa é por ela se tratar de soluções existentes para lidar com Requisitos de Qualidade negligenciados onde a maioria das soluções propostas são criações de novos artefatos. Essa

comparação visa identificar linhas de pesquisas que poderão beneficiar o mercado de desenvolvimento de software com métodos ágeis.

Este trabalho é composto por 5 seções, incluindo esta seção de introdução. A seção 2 é voltada à revisão bibliográfica que concerne este trabalho, no qual traz os principais conceitos de engenharia de requisitos, requisitos de qualidade, metodologias ágeis, além dos principais resultados encontrados na RSL dos autores Alsaqaf, Daneva & Wieringa (2017). A seção 3 contempla o procedimento metodológico adotado nesta pesquisa, com a classificação da mesma, o questionário utilizado para colher insumos para este trabalho, uma seção voltada às etapas da pesquisa e, por fim, as considerações finais com relação à metodologia adotada. A seção 4 traz a análise dos resultados encontrados no survey comparados com duas das três respostas das perguntas de pesquisa presentes na RSL dos autores Alsaqaf, Daneva & Wieringa (2017). A seção expressa as características gerais da amostra e as considerações finais sobre a análise dos resultados. Por fim, a seção 5 abrange a conclusão deste trabalho, como também suas contribuições para o mercado e a academia, determina as limitações encontradas em sua execução e propõe sugestões de trabalhos futuros voltados ao assunto abordado.

## **2 Revisão bibliográfica**

Esta seção resume os principais conceitos que estão relacionados com a prática da Engenharia de Requisitos, os Requisitos de Qualidade e as Metodologias Ágeis mais utilizadas no mercado. Além disso, apresenta uma explanação sobre os resultados da RSL dos autores Alsaqaf, Daneva & Wieringa (2017), que serviu como referência para esta pesquisa.

### **2.1 Engenharia de requisitos**

O Institute of Electrical and Electronics Engineers (IEEE, 1984) define a Engenharia de Requisitos como o processo de aquisição, refinamento e verificação das necessidades do cliente para um sistema de software, objetivando-se ter uma especificação completa e correta dos requisitos de software.

Já na visão de Sommerville (2011, p.24), engenharia de requisitos é o processo de compreensão e definição dos serviços requisitados do sistema e identificação de restrições relativas à operação e ao desenvolvimento do sistema. A engenharia de requisitos é um estágio particularmente crítico do processo de software, pois erros nessa fase inevitavelmente geram problemas no projeto e na implementação do sistema.

O processo de engenharia de requisitos (Figura 1) tem como objetivo produzir um documento de requisitos acordados que especifica um sistema que satisfaz os requisitos dos stakeholders. Requisitos são geralmente apresentados em dois níveis de detalhe. Os usuários finais e os clientes precisam de uma declaração de requisitos em alto nível; desenvolvedores de sistemas precisam de uma especificação mais detalhada do sistema.

Existem quatro atividades principais do processo de engenharia de requisitos:

1. *Estudo de viabilidade.* É feita uma estimativa acerca da possibilidade de se satisfazerem as necessidades do usuário identificado usando-se tecnologias atuais de software e hardware. O estudo considera se o sistema proposto será rentável a partir de um ponto de vista de negócio e se ele pode ser desenvolvido no âmbito das atuais restrições orçamentais. Um estudo de viabilidade deve ser relativamente barato e rápido. O resultado deve informar a decisão de avançar ou não, com uma análise mais detalhada.

2. *Elicitação e análise de requisitos.* Esse é o processo de derivação dos requisitos do sistema por meio da observação dos sistemas existentes, além de discussões com os potenciais usuários e compradores, análise de tarefas, entre outras etapas. Essa parte do processo pode envolver o desenvolvimento de um ou mais modelos de sistemas e protótipos, os quais nos ajudam a entender o sistema a ser especificado.

3. *Especificação de requisitos.* É a atividade de traduzir as informações obtidas durante a atividade de análise em um documento que define um conjunto de requisitos. Dois tipos de requisitos podem ser incluídos nesse documento. **Requisitos do usuário** são declarações abstratas dos requisitos do sistema para o cliente e usuário final do sistema; **Requisitos de sistema** são descrições mais detalhadas da funcionalidade a ser provida.

4. *A validação de requisitos.* Essa atividade verifica os requisitos quanto ao realismo, consistência e completude. Durante esse processo, os erros no documento de requisitos são inevitavelmente descobertos. Em seguida, o documento deve ser modificado para correção desses problemas.

Naturalmente, as atividades no processo de requisitos não são feitas em apenas uma sequência. A análise de requisitos continua durante a definição e especificação, e novos requisitos emergem durante o processo. Portanto, as atividades de análise, definição e especificação são intercaladas. Nos métodos ágeis, como extreme programming, os requisitos são desenvolvidos de forma incremental, de acordo com as prioridades do usuário, e a elicitação de requisitos é feita pelos usuários que integram a equipe de desenvolvimento.

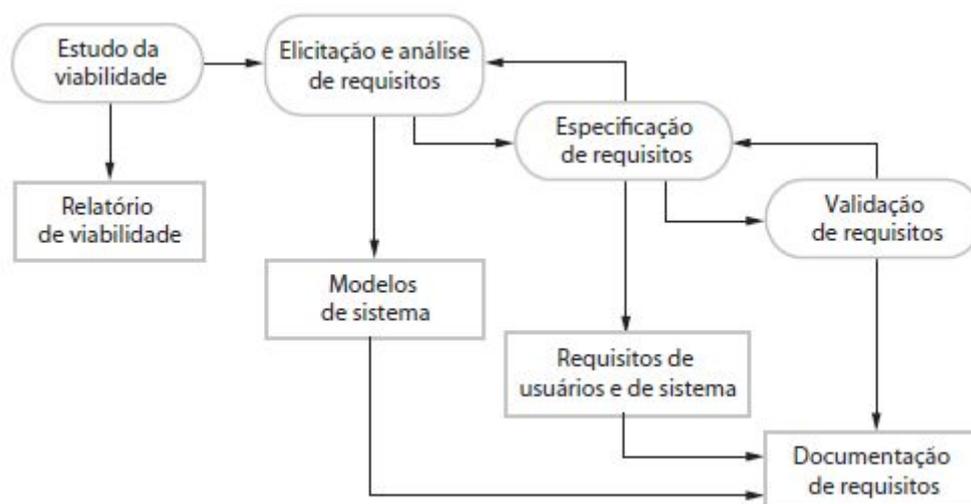


Figura 1: Os requisitos da engenharia de processos  
(Fonte: SOMMERVILLE, 2011)

Os requisitos de software são frequentemente classificados como **requisitos funcionais e requisitos não funcionais** (SOMMERVILLE, 2011):

1. *Requisitos funcionais*. São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer.

2. *Requisitos não funcionais*. São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo.

## 2.2 Requisitos de qualidade

De acordo com SOMMERVILLE (2011, p.60), os requisitos não funcionais, como o nome sugere, são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área. Uma alternativa a esse cenário seria os requisitos definirem restrições sobre a implementação do sistema, como as capacidades dos dispositivos de E/S ou as representações de dados usadas nas interfaces com outros sistemas. Os requisitos não funcionais, como desempenho, proteção ou disponibilidade, normalmente especificam ou restringem as características do sistema como um todo. Requisitos não funcionais são frequentemente mais críticos que requisitos funcionais individuais. Os usuários do sistema podem, geralmente, encontrar maneiras de contornar uma função do sistema que realmente não atenda a suas necessidades. No entanto, deixar de atender a um requisito não funcional pode significar a inutilização de todo o sistema.

Já para Mairiza; Zowghi; Nurmuliani (2010, p.311), os Requisitos Não-Funcionais (NFRs) são reconhecidos como fator muito importante para o sucesso do projeto de software. Se os NFRs não forem abordados adequadamente, vários problemas potenciais podem ocorrer. Por exemplo, software que é inconsistente e de baixa qualidade; insatisfação de clientes, usuários finais e desenvolvedores; causando sobrecarga de tempo e custo para correção de erros de software. No ciclo de vida de desenvolvimento de software, os NFRs são considerados como restrições ou qualificações das operações.

Embora os NFRs sejam amplamente reconhecidos como muito importantes, a revisão da literatura mostra que os NFRs são frequentemente negligenciados, mal compreendidos e não considerados adequados no desenvolvimento de software. No desenvolvimento de sistema de software, os usuários focam na especificação dos requisitos funcionais ou comportamentais do sistema, isto é, coisas que o produto deve fazer (Wagner et al., 2018).

A Figura 2 mostra o resultado da investigação sobre definição e terminologia dos Requisitos Não-Funcionais na literatura. Esta figura ilustra que geralmente, o termo NFRs é considerado para duas perspectivas: (1) NFRs como os requisitos que descrevem as propriedades, características ou restrições que um sistema de software deve exibir; e (2) NFRs

como os requisitos que descrevem o atributos de qualidade que o produto de software deve ter. Na primeira perspectiva, os NFRs consistem em vários aspectos, como restrições de desenvolvimento, regras de negócios, interfaces externas, atributos de qualidade e quaisquer outros requisitos que não descrevam a funcionalidade do sistema. O termo restrições, requisitos não comportamentais, preocupações, objetivos e requisitos extra-funcionais também são usados para representar os NFRs nessa perspectiva. A segunda perspectiva leva o estreito foco dos NFRs considerando os atributos de qualidade. Portanto, essa perspectiva é o subconjunto da primeira perspectiva. O termo requisitos de qualidade, atributos do sistema de software e atributos de qualidade também são usados para representar os NFRs.

Requisitos de qualidade (QRs) são os requisitos que descrevem as qualidades do sistema (por exemplo, requisitos de desempenho, requisitos de manutenção). Na literatura de engenharia de requisitos (RE), há um consenso de que o sucesso ou o fracasso de um sistema não é decidido apenas pela implementação dos requisitos funcionais corretos, mas também pela correta implementação dos QRs. Por exemplo, se o tempo de resposta do sistema não atender às expectativas do cliente, não podemos dizer que o sistema oferece qualidade. (ALSAQAF; DANEVA; WIERINGA, 2017).

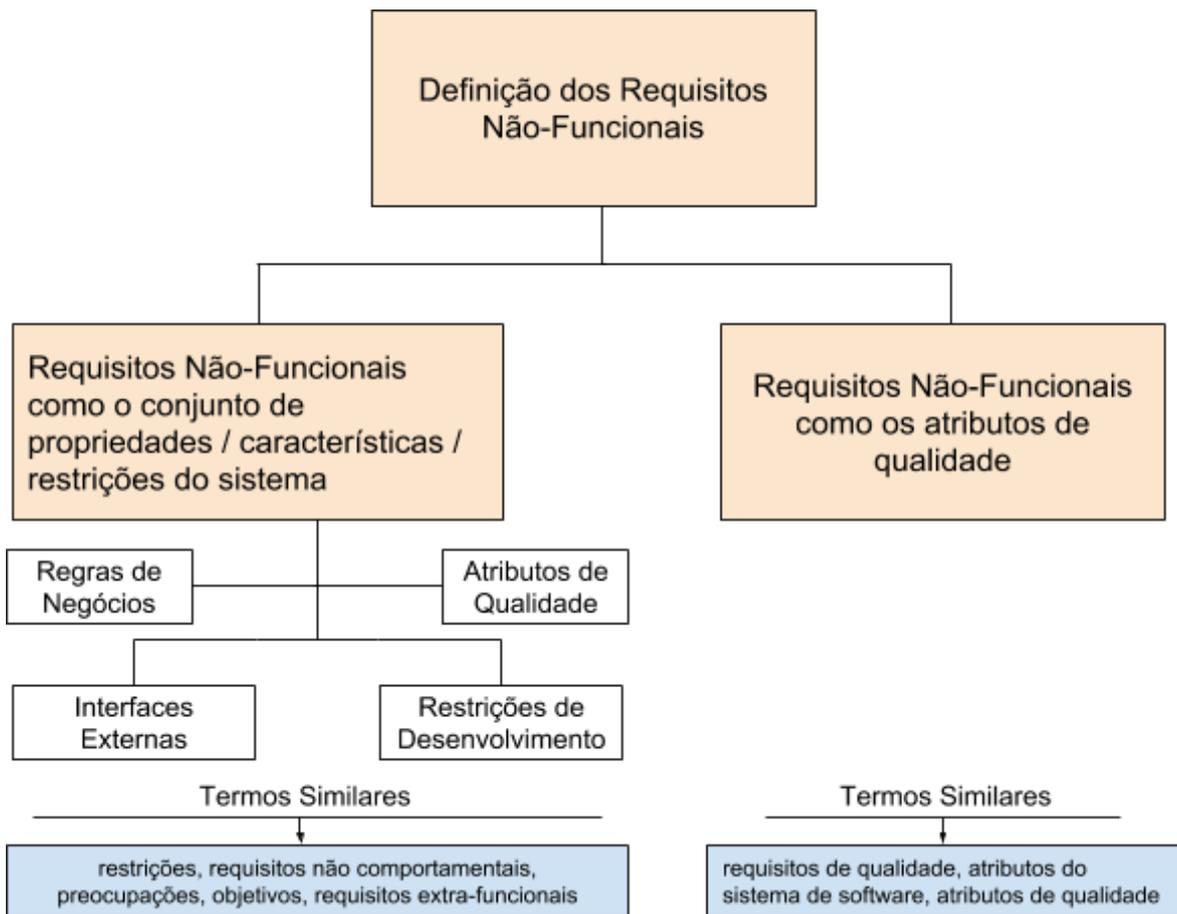


Figura 2: Definição e terminologia dos Requisitos Não-Funcionais  
(Fonte:MAIRIZA; ZOWGHI; NURMULIANI, 2010)

## 2.3 Metodologias ágeis

As metodologias ágeis de desenvolvimento de software, ou Agile, surgiram e ganharam muita força nos últimos anos devido à necessidade do mercado em atender às demandas dos clientes e seus projetos de maneira mais dinâmica, flexível e com maior produtividade.

O desenvolvimento ágil utiliza uma abordagem de planejamento incremental e muito iterativa. Dessa forma, diferentemente do método tradicional em que a princípio se define e se documenta detalhadamente todas as fases do início ao fim do projeto, no método ágil isso é feito em pequenas partes, também chamadas iterações.

Cada iteração é um mini-projeto, que normalmente dura de 1 a 4 semanas, e inclui todas as fases para implementá-lo como levantamento de recursos e requisitos, projeto, desenvolvimento de código, testes e documentação. Ao final de cada iteração deve haver uma entrega ao cliente, que inclua um conjunto de novas funcionalidades, uma nova versão de software. Após essa entrega há um novo processo de comunicação com o cliente e então são definidas quais deverão ser as novas entregas.

Esse processo visa que o cliente receba regularmente uma nova porção funcional de seu produto, visa também uma proximidade e colaboração constante entre o cliente e a equipe de desenvolvimento e entrega, reduzindo riscos de falhas de comunicação e possibilitando mudanças mais ágeis no andamento do projeto. Dessa forma, o cliente recebe um produto realmente útil à sua empresa e com maior qualidade (METODOLOGIA Ágil, Dinamismo no Desenvolvimento de Software [2016]).

Segundo Sommerville (2011, p.40), a filosofia por trás dos métodos ágeis é refletida no manifesto ágil, que foi acordado por muitos dos principais desenvolvedores desses métodos. Esse manifesto afirma: Estamos descobrindo melhores maneiras de desenvolver softwares, fazendo-o e ajudando outros a fazê-lo. Através desse trabalho, valorizamos mais:

- Indivíduos e interações do que processos e ferramentas;
- Software em funcionamento do que documentação abrangente;
- Colaboração do cliente do que negociação de contrato;
- Respostas a mudanças do que seguir um plano;

Ou seja, embora itens à direita sejam importantes, valorizamos mais os que estão à esquerda. Embora esses métodos ágeis sejam todos baseados na noção de desenvolvimento e entrega incremental, eles propõem diferentes processos para alcançar tal objetivo. No entanto, compartilham um conjunto de princípios, com base no manifesto ágil, e por isso têm muito em comum. Esses princípios são mostrados na Figura 3.

Princípios	Descrição
Envolvimento do cliente	Os clientes devem estar intimamente envolvidos no processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos do sistema e avaliar suas iterações.
Entrega incremental	O software é desenvolvido em incrementos com o cliente, especificando os requisitos para serem incluídos em cada um.
Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Membros da equipe devem desenvolver suas próprias maneiras de trabalhar, sem processos prescritivos.
Aceitar as mudanças	Deve-se ter em mente que os requisitos do sistema vão mudar. Por isso, projete o sistema de maneira a acomodar essas mudanças.
Manter a simplicidade	Focalize a simplicidade, tanto do software a ser desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.

Figura 3: Os princípios dos métodos ágeis  
(Fonte: SOMMERVILLE, 2011)

De acordo com (METODOLOGIA Ágil, Dinamismo no Desenvolvimento de Software [2016]), para seguir os valores e princípios do manifesto ágil, as diversas metodologias ágeis costumam utilizar as seguintes técnicas e práticas durante todo o processo:

- *Equipes Auto Gerenciáveis*: É importante que a equipe atue em conjunto para tomar as melhores decisões, ao invés de haver um gerente que dita o caminho, há um líder que atua como facilitador;
- *Comunicação*: É essencial a constante comunicação entre a equipe de desenvolvimento e a área de negócio;
- *Preferência por Equipes Pequenas*: É consenso que equipes pequenas são mais produtivas, evitam conflitos e falhas de comunicação;
- *Uso de TDD (Test-Driven Development)*: É uma técnica que permite a realização de testes contínuos e não apenas na conclusão do sistema;
- *Planejamento Incremental*: Ao invés de planejar o software como um todo, o planejamento é feito de forma sistêmica. O todo é determinado, mas o planejamento é feito por etapas;
- *Entregas menores*: Uso de incrementos pequenos de software, entregando novas funcionalidades em meses ou semanas, ao invés de anos;
- *Uso de Refatoração*: Melhoria constante do código, tornando-o mais fácil de manter;
- *Integração sistêmica contínua*: Sempre que um incremento está pronto, ele é integrado ao sistema como um todo.

Algumas das **principais vantagens do uso dos métodos ágeis de desenvolvimento de software**, tanto do lado dos clientes, quando das equipes de desenvolvimento e entrega são:

*Para o cliente:*

1. Mais entrega de Valor ao Negócio, e conseqüente um melhor ROI (Return On Investment);
2. Produto final com melhor qualidade;

3. Entregas mais frequentes e regulares, os usuários podem ver os resultados parciais do projeto;
4. Melhor Time-to-Market gerando um ganho de competitividade;
5. Melhor Transparência e Visibilidade da situação do projeto;
6. Maior flexibilidade a mudanças e ajustes nos requisitos e nas prioridades de entrega.

*Para a área de desenvolvimento:*

1. Melhor comunicação com o cliente e usuários finais;
2. Antecipação de Problemas e tomadas de decisão mais rápidas;
3. Definição de escopo, objetivos e prioridades mais claros;
4. Maior autonomia e responsabilidade das equipes, gerando conseqüentemente mais comprometimento;
5. Melhoria contínua do processo de desenvolvimento.

Em 2001 foi publicado, a partir do trabalho conjunto de diversos especialistas da área de sistemas, o Manifesto para Desenvolvimento Ágil de Software. Tal declaração enfatiza a entrega de software operável, atuando em conjunto com o solicitante do mesmo, além de frisar a necessidade de uma interação adequada entre os diversos envolvidos num projeto e a flexibilidade diante de mudanças. São doze os princípios que norteiam o Manifesto Ágil (ou em inglês “Agile Manifesto”) (GROFFE, 2018).

De acordo com BECK et al., (2018) os doze princípios por trás do manifesto ágil são:

1. Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
3. Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
4. Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
5. Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
6. O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
7. Software funcional é a medida primária de progresso.
8. Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
9. Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
10. Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
11. As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.

12. Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

Segundo Totvs (2018), as metodologias ágeis são alternativas ao modelo tradicional de gestão de projeto. O objetivo é auxiliar as empresas a lidarem com imprevisibilidades dentro de projetos, por intermédio de ciclos que estimulam maior interação entre consumidores e colaboradores. Existem entregas parciais dinâmicas e constantes, isto é, o cliente não necessita aguardar muito tempo para acessar resultados. As metodologias ágeis ainda estimulam a adaptação contínua e as atuações em equipe. Foco no consumidor, produção de maior valor agregado e auto-organização integram os seus pilares.

Suas práticas têm por propósito a entrega rápida e de alta qualidade do produto final do projeto, permitindo ainda que seja entregue em partes à medida que se desenvolve. Basicamente, as abordagens das metodologias ágeis sintonizam a criação e a condução de projetos aos objetivos do negócio e às necessidades dos consumidores. Esses princípios geram benefícios às empresas, como aumento na satisfação do público e comunicação ativa entre os participantes do projeto e os clientes. Além do mais, há grande enfoque em prazos, de modo que é possível sincronizar o cronograma de execução de orçamento do projeto com as etapas e entregas aos clientes.

Com base nos conhecimentos expostos sobre Metodologias Ágeis, podemos então abordar os conceitos dos dois processos ágeis mais utilizados no mercado atualmente.

### 2.3.1 XP (Extreme Programming)

De acordo com Sommerville (2011, p.44), o nome foi cunhado por Beck (2000), pois a abordagem foi desenvolvida para impulsionar práticas reconhecidamente boas, como o desenvolvimento iterativo, a níveis ‘extremos’. Por exemplo, em XP, várias novas versões de um sistema podem ser desenvolvidas, integradas e testadas em um único dia por programadores diferentes.

Em Extreme Programming, os requisitos são expressos como cenários (chamados de histórias do usuário), que são implementados diretamente como uma série de tarefas. Os programadores trabalham em pares e desenvolvem testes para cada tarefa antes de escreverem o código. Quando o novo código é integrado ao sistema, todos os testes devem ser executados com sucesso. Há um curto intervalo entre os releases do sistema. A Figura 4 ilustra o processo XP para a produção de um incremento do sistema que está sendo desenvolvido.

Extreme Programming envolve uma série de práticas que refletem os princípios dos métodos ágeis (elas estão resumidas na Tabela 1):

1. O desenvolvimento incremental é sustentado por meio de pequenos e frequentes releases do sistema. Os requisitos são baseados em cenários ou em simples histórias de clientes, usadas como base para decidir a funcionalidade que deve ser incluída em um incremento do sistema.

2. O envolvimento do cliente é sustentado por meio do engajamento contínuo do cliente com a equipe de desenvolvimento. O representante do cliente participa do desenvolvimento e é responsável por definir os testes de aceitação para o sistema.
3. Pessoas — não processos — são sustentadas por meio de programação em pares, propriedade coletiva do código do sistema e um processo de desenvolvimento sustentável que não envolve horas de trabalho excessivamente longas.
4. As mudanças são aceitas por meio de releases contínuos para os clientes, do desenvolvimento test-first, da refatoração para evitar a degeneração do código e integração contínua de nova funcionalidade.
5. A manutenção da simplicidade é feita por meio da refatoração constante que melhora a qualidade do código, bem como por meio de projetos simples que não antecipam desnecessariamente futuras mudanças no sistema.

Em um processo XP, os clientes estão intimamente envolvidos na especificação e priorização dos requisitos do sistema. Os requisitos não estão especificados como uma lista de funções requeridas do sistema. Pelo contrário, o cliente do sistema é parte da equipe de desenvolvimento e discute cenários com outros membros da equipe. Juntos, eles desenvolvem um ‘cartão de estória’, englobando as necessidades do cliente. A equipe de desenvolvimento, então, tenta implementar esse cenário em um release futuro do software.

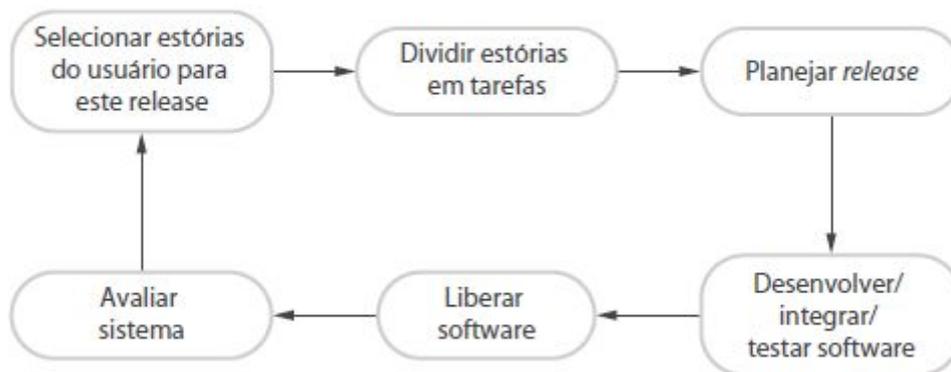


Figura 4: O ciclo de um release em Extreme Programming  
(Fonte: SOMMERVILLE, 2011)

Princípio ou prática	Descrição
Planejamento incremental	Os requisitos são gravados em cartões de estória e as estórias que serão incluídas em um release são determinadas pelo tempo disponível e sua relativa prioridade. Os desenvolvedores dividem essas estórias em 'Tarefas'. Veja os quadros 3.1 e 3.2.
Pequenos <i>releases</i>	Em primeiro lugar, desenvolve-se um conjunto mínimo de funcionalidades útil, que fornece o valor do negócio. <i>Releases</i> do sistema são frequentes e gradualmente adicionam funcionalidade ao primeiro <i>release</i> .
Projeto simples	Cada projeto é realizado para atender às necessidades atuais, e nada mais.
Desenvolvimento <i>test-first</i>	Um <i>framework</i> de testes iniciais automatizados é usado para escrever os testes para uma nova funcionalidade antes que a funcionalidade em si seja implementada.
Refatoração	Todos os desenvolvedores devem refatorar o código continuamente assim que encontrarem melhorias de código. Isso mantém o código simples e manutenível.
Programação em pares	Os desenvolvedores trabalham em pares, verificando o trabalho dos outros e prestando apoio para um bom trabalho sempre.
Propriedade coletiva	Os pares de desenvolvedores trabalham em todas as áreas do sistema, de modo que não se desenvolvam ilhas de <i>expertise</i> . Todos os conhecimentos e todos os desenvolvedores assumem responsabilidade por todo o código. Qualquer um pode mudar qualquer coisa.
Integração contínua	Assim que o trabalho em uma tarefa é concluído, ele é integrado ao sistema como um todo. Após essa integração, todos os testes de unidade do sistema devem passar.
Ritmo sustentável	Grandes quantidades de horas-extra não são consideradas aceitáveis, pois o resultado final, muitas vezes, é a redução da qualidade do código e da produtividade a médio prazo.
Cliente no local	Um representante do usuário final do sistema (o cliente) deve estar disponível todo o tempo à equipe de XP. Em um processo de Extreme Programming, o cliente é um membro da equipe de desenvolvimento e é responsável por levar a ela os requisitos de sistema para implementação.

Tabela 1: Práticas de Extreme Programming  
(Fonte: SOMMERVILLE, 2011)

### 2.3.2 Scrum

Segundo Sommerville (2011, p.50), a abordagem Scrum é um método ágil geral, mas seu foco está no gerenciamento do desenvolvimento iterativo, ao invés das abordagens técnicas específicas da engenharia de software ágil. A Figura 5 é um diagrama do processo Scrum de gerenciamento. Scrum não prescreve o uso de práticas de programação, como programação em pares e desenvolvimento test-first.

No Scrum, existem três fases. A primeira é uma fase de planejamento geral, em que se estabelecem os objetivos gerais do projeto e da arquitetura do software. Em seguida, ocorre uma série de ciclos de sprint, sendo que cada ciclo desenvolve um incremento do sistema. Finalmente, a última fase do projeto encerra o projeto, completa a documentação exigida, como quadros de ajuda do sistema e manuais do usuário, e avalia as lições aprendidas com o projeto.

A característica inovadora do Scrum é sua fase central, chamada ciclos de sprint. Um sprint do Scrum é uma unidade de planejamento na qual o trabalho a ser feito é avaliado, os recursos para o desenvolvimento são selecionados e o software é implementado. No fim de um sprint, a funcionalidade completa é entregue aos stakeholders.

As principais características desse processo são:

1. Sprints são de comprimento fixo, normalmente duas a quatro semanas. Eles correspondem ao desenvolvimento de um release do sistema em XP.
2. O ponto de partida para o planejamento é o backlog do produto, que é a lista do trabalho a ser feito no projeto. Durante a fase de avaliação do sprint, este é revisto, e as prioridades e os riscos são identificados. O cliente está intimamente envolvido nesse processo e, no início de cada sprint, pode introduzir novos requisitos ou tarefas.
3. A fase de seleção envolve todos da equipe do projeto que trabalham com o cliente para selecionar os recursos e a funcionalidade a ser desenvolvida durante o sprint.
4. Uma vez que todos estejam de acordo, a equipe se organiza para desenvolver o software. Reuniões diárias rápidas, envolvendo todos os membros da equipe, são realizadas para analisar os progressos e, se necessário, repriorizar o trabalho. Nessa etapa, a equipe está isolada do cliente e da organização, com todas as comunicações canalizadas por meio do chamado ‘Scrum Master’. O papel do Scrum Master é proteger a equipe de desenvolvimento de distrações externas. A maneira como o trabalho é desenvolvido depende do problema e da equipe. Diferentemente do XP, a abordagem Scrum não faz sugestões específicas sobre como escrever os requisitos ou sobre o desenvolvimento test-first etc. No entanto, essas práticas de XP podem ser usadas se a equipe achar que são adequadas.
5. No fim do sprint, o trabalho é revisto e apresentado aos stakeholders. O próximo ciclo sprint começa em seguida.

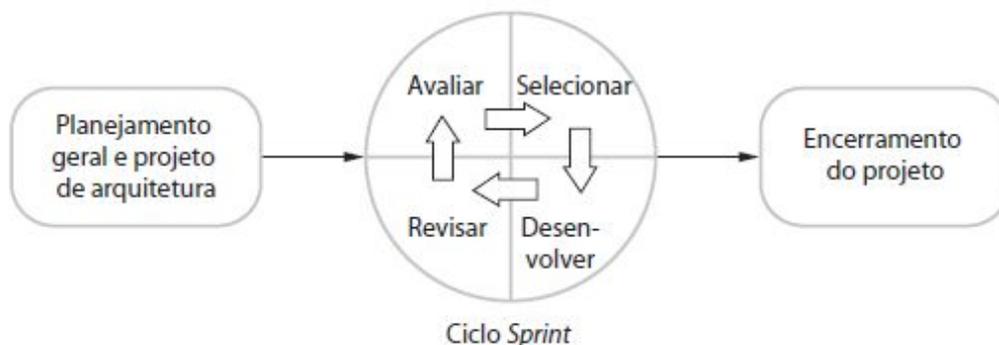


Figura 5: O processo Scrum  
(Fonte: SOMMERVILLE, 2011)

#### 2.4 Resultados de uma revisão sistemática da literatura

A revisão sistemática de literatura intitulada de “Quality Requirements in Large-Scale Distributed Agile Projects – A Systematic Literature Review”, foi motivada pela percepção dos autores que os métodos de desenvolvimento ágeis tornaram-se cada vez mais popular nos últimos anos. No entanto, esses métodos dificilmente atentam para os requisitos de qualidade (QRs), que poderia prejudicar os lucros da entrega rápida através da introdução de altos esforços de retrabalho posteriormente. Segundo os autores, já existem várias publicações que

relatam a atenção insuficiente aos requisitos de qualidade em métodos ágeis, mesmo conhecendo as práticas ágeis de engenharia de requisitos e seu impacto nos requisitos de qualidade em ambientes distribuídos e de grande escala, as empresas usam cada vez mais os métodos ágeis nesses ambientes, tendo como resultado negativo ignorar os requisitos de qualidade cada vez mais. Com isso, a pesquisa teve como objetivo identificar os desafios na engenharia de requisitos ágeis de qualidade distribuída em projetos de grande escala que foram pesquisados até agora, as práticas ágeis que contribuíram para o surgimento desses desafios e as soluções propostas. Nessa revisão sistemática foram encontrados 60 estudos relevantes que, conseqüentemente, foram analisados e sintetizados.

O estudo foi realizado com base nas diretrizes de Kitchenham et al., para que os resultados obtidos fossem confiáveis e auditáveis, além de haver a possibilidade dos mesmos serem reproduzidos por outros pesquisadores.

Como indicado anteriormente o enfoque da revisão está nas práticas de projetos ágeis distribuídos em grande escala e nas abordagens propostas para lidar com os desafios dos requisitos de qualidade nesses projetos. Para que fosse atingido os objetivos estipulados para a pesquisa, foram criadas três questões de pesquisa:

- RQ1: Quais são as práticas ágeis usadas para projetar Requisitos de Qualidade em ambientes de Larga Escala, de acordo com a literatura publicada?
- RQ2: Quais desafios de Requisitos de Qualidade foram relatados em projetos ágeis, em geral?
- RQ3: Quais são as soluções existentes para lidar com Requisitos de Qualidade negligenciados em Engenharia de Requisitos Ágeis em geral (não apenas em Larga Escala), conforme a literatura de Engenharia de Requisitos?

Segundo os autores, para evitar fazer um trabalho que já foi feito, foram procuradas Revisões Sistemáticas já existentes que poderiam responder às três questões de pesquisa. Para coletar os documentos, optaram por usar o engenho de busca do Scopus. Depois de concluir que as perguntas ainda não tinham sido respondidas, os autores começaram a extrair palavras-chave das questões de pesquisas elaboradas. As palavras-chave relacionadas a ambientes distribuídos e de larga escala acabaram removidas da string de busca porque não queriam perder nenhum conhecimento que poderia ser perdido se a pesquisa fosse restringida a um ambiente específico. A string final foi:“(Agile OR agility) AND (Requirements OR non-functional requirements OR non-functional OR quality requirements OR quality attributes OR quality)”.

Na visão geral dos estudos, foram retornados um grande número de publicações: 2830. Para reduzir este número a um tamanho gerenciável, os autores limitaram o assunto da pesquisa à 'Ciência da Computação' no Scopus, que resultou em 612 artigos. O primeiro autor leu então os resumos desses artigos para aplicar os critérios de inclusão e exclusão. Ele removeu manualmente 446 papéis. As razões para a remoção desses artigos, foram que esses artigos ou não atendiam ao critério de inclusão I3 (O artigo discute dois ou mais dos seguintes tópicos: Agile, RE, QRs, projetos distribuídos e de larga escala). Ou atendiam ao critério de exclusão E2 (O artigo não discute Engenharia de Requisitos como seu tema central). A

aplicação dos critérios de inclusão e exclusão reduziu o número de artigos para 166. O segundo autor verificou os artigos removidos separadamente para certificar de que nenhum documento foi removido por engano. O número de artigos permaneceu intacto. Depois desta etapa, os dois primeiros autores leram e analisaram o título, palavras-chave, resumo, introdução e conclusão de todos os 166 trabalhos, a fim de avaliar a sua relevância para os tópicos de interesse. Os autores consideraram um artigo 'relevante' se ele fornece informações “suficientes” para responder a uma ou mais das questões de pesquisa. Isso totalizou 51 trabalhos. Outros 9 artigos foram adicionados aos artigos primários selecionados com base no 'efeito bola de neve', que elevou o número final de artigos primários para 60.

Os 60 artigos incluídos neste estudo foram distribuídos de acordo com as categorias de publicação, conforme Figura 6.

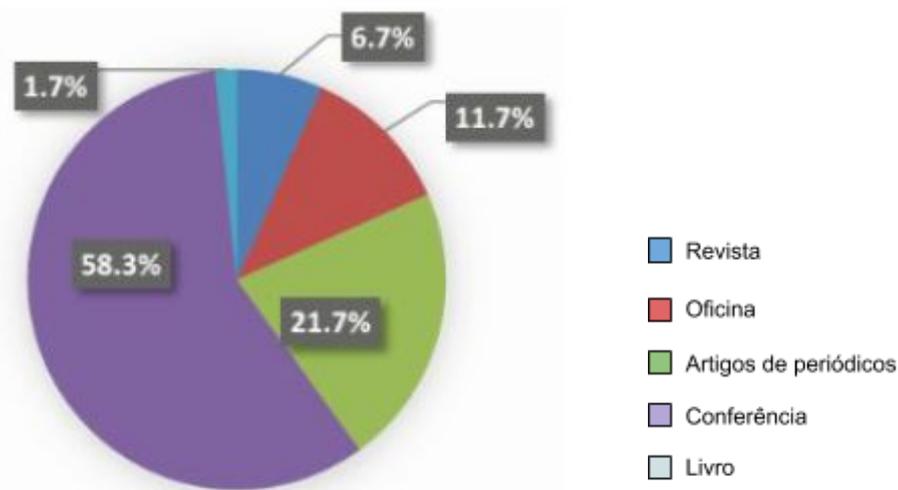


Figura 6: Distribuição dos artigos de acordo com as categorias de publicação  
(Fonte: ALSAQAF; DANEVA; WIERINGA, 2017)

A conferência que publicou o maior número de artigos é a Conferência SoutheastCon ([www.ewh.ieee.org/reg/3/southeastcon/](http://www.ewh.ieee.org/reg/3/southeastcon/)) com 6 publicações. As conferências internacionais sobre Engenharia de Requisitos (ER) e a Engenharia Global de Software publicaram 3 trabalhos cada. A conferência de engenharia de software da Malásia publicou 2 trabalhos, enquanto todos os outros locais de conferência (21 conferências no total) são representadas por uma publicação cada. Em relação a revistas, o IEEE Software foi o único que publicou trabalhos incluídos nesta Revisão Sistemática (3 no total). Em termos de artigos de periódicos, o Jornal de Sistemas e Software foi o que mais publicou (3 artigos), seguido por Informação e Tecnologia de Software (2 trabalhos).

Em relação ao ano de publicação, foram constatados que 53 artigos foram publicados após 2008, enquanto 7 artigos foram publicados no período de 2002-2008. Isso indica que o

tópico deste estudo ganhou mais atenção nos últimos sete anos. 2015 foi o ano mais produtivo com 11 artigos publicados (Figura 7).

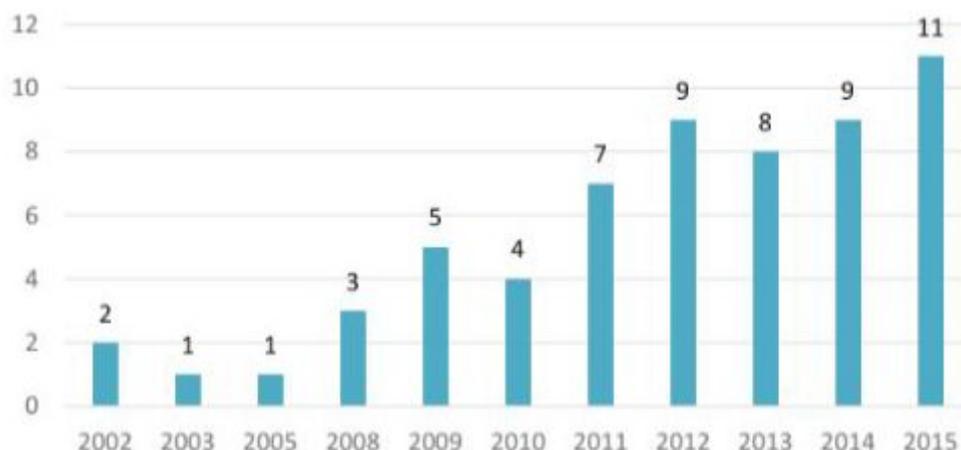


Figura 7: Número de publicações por ano  
(Fonte: ALSAQAF; DANEVA; WIERINGA, 2017)

Em resposta à primeira questão de pesquisa, que questionava quais práticas ágeis eram usadas para projetar Requisitos de Qualidade em ambientes de Larga Escala, de acordo com a literatura publicada, mesmo sem a intenção dos autores foram encontrados artigos que analisaram quatro tipos de Requisitos de Qualidade: Segurança, Safety, Conformidade e Usabilidade. Cada um desses artigos enfocou as práticas específicas atualmente usadas para projetar os respectivos tipos de Requisitos de Qualidade discutidos no trabalho, independentemente de outros Requisitos de Qualidade. Em essência, essas práticas específicas sugerem a introdução de um novo artefato em Engenharia de Requisitos Ágeis, ou um novo papel, ou novo comportamento social, ou uma nova ferramenta, ou uma combinação de todos. Um resumo dessas práticas pode ser encontrado na Tabela 2, onde a coluna mais à direita da tabela indica se a prática diz respeito a artefatos, ferramentas, papéis ou comportamento social

Prática	Descrição	Referências	Tipo de Prática
Comunicação face a face	Sessões de comunicação entre a equipe de desenvolvimento e o Dono/Representante do Produto (Product Owner) para identificar os requisitos	P2, P4, P12, P13, P14, P15, P16	Baseado em Comportamento social
Emergência iterativa de requisitos	Os requisitos emergem durante o processo de desenvolvimento em vez de serem definidos antecipadamente	P2, P4, P12, P17, P18	Baseado em Artefato

Priorização frequente de requisitos	Prioriza continuamente as histórias do usuário com base nas mudanças dos valores do negócio	P2, P19	Baseado em Artefato
Histórias de usuários	Descrição breve e abstrata dos requisitos usados para incentivar a comunicação face a face	P2, P4, P14, P17, P20, P21, P22, P23, P24, P25	Baseado em Artefato
Preparação de produtos	Revisita continuamente o conjunto de histórias de usuários para remover histórias obsoletas, criar novas histórias e reavaliar sua prioridade relativa	P14	Baseado em Artefato
História de entrega	História do usuário estendida com informações adicionais, tais como: especificação funcional, requisitos de qualidade, design de alto nível e cenários de teste	P19	Baseado em Artefato
Wiki	Ferramenta on-line que pode ser acessada por todos os membros da equipe para comunicar as histórias do usuário	P24, P26	Baseado em Ferramenta
Prototipação evolutiva	Um protótipo que é construído iterativamente para ser finalmente usado na produção	P2, P4	Baseado em Artefato
Desenvolvimento orientado a testes (TDD)	Os requisitos são primeiramente transformados em casos de teste e, em seguida, o código necessário é gravado para passar nos casos de teste	P2, P14, P28	Baseado em Artefato
Integração contínua	Todas as cópias dos produtos de todos os membros da equipe são mescladas várias vezes ao dia para validar o funcionamento correto do produto	P14, P28, P29	Baseado em Artefato
Refatoração	Pouco antes do final de cada	P14	Baseado em

	iteração, a estrutura interna do produto é reestruturada para melhorar a qualidade interna		Artefato
Programação em pares	Dois desenvolvedores trabalham juntos. Uma pessoa é responsável pela programação e a outra é responsável por controlar e validar o código	P14	Comportamento social, baseado em artefatos

Tabela 2: Práticas ágeis sobre Engenharia de Requisitos em ambientes de Larga Escala  
(Fonte: ALSAQAF; DANEVA; WIERINGA, 2017)

Com relação à segunda questão de pesquisa, os pesquisadores buscaram saber os desafios de Requisitos de Qualidade que foram relatados em projetos ágeis, em geral. Vinte e seis artigos relataram explicitamente e implicitamente a negligência dos Requisitos de Qualidade em projetos ágeis. A Tabela 3 resume 12 desafios específicos causados por práticas ágeis em relação aos Requisitos de Qualidade em geral, de acordo com esses artigos. A Tabela 3 sugere que algumas características de Engenharia de Requisitos Ágeis são fortes em livros didáticos ágeis (por exemplo, o papel do product owner, o uso de histórias de usuários) pode ser considerado de fato como inibidor na engenharia de requisitos de qualidade. Quatro dos 12 desafios estão relacionados ao papel do product owner. A este respeito, 11 artigos apontam para vários aspectos associados a esse papel como condutores de dificuldades de Requisitos de Qualidades, por conhecimento insuficiente sobre o lado da qualidade do produto em relação aos Requisitos de Qualidade, sendo sobrecarregado único ponto de contato em Requisitos de Qualidade. Isso mostra que o Product Owner é uma fonte potencial de desafios de Engenharia de Requisitos.

Desafios	Referências
Métodos ágeis não fornecem uma técnica amplamente aceita para reunir os Requisitos de Qualidade.	P32, P34, P37
Foco no fornecimento de funcionalidade ao custo da flexibilidade da arquitetura.	P2, P20, P31, P36, P38, P39, P40
Ignorar os requisitos de arquitetura previsíveis.	P38, P39
Incapacidade de histórias de usuários para documentar os Requisitos de Qualidade.	P19, P21, P35, P36
Incapacidade de histórias do usuário para documentar dependências de requisitos.	P19, P34, P36
A validação dos Requisitos de Qualidade ocorre tarde demais	P4, P19, P10, P27, P41

no processo.	
Falta de mecanismo de rastreabilidade de requisitos.	P10, P30, P32, P33, P42, P43, P44
Falta de conhecimento específico do proprietário/representante do produto (Product Owner).	P14, P16, P19, P36, P38
Sobrecarga de trabalho do proprietário/representante do produto (Product Owner).	P14, P16
Disponibilidade insuficiente do proprietário/representante do produto (Product Owner).	P17
Análise insuficiente de requisitos.	P40, P45
Dependência do proprietário/representante do produto (Product Owner) como ponto único para coletar os requisitos.	P33, P38, P40

Tabela 3: Desafios relatados em relação aos Requisitos de Qualidade  
(Fonte: ALSAQAF; DANEVA; WIERINGA, 2017)

Por fim, os pesquisadores buscaram entender, através da terceira questão de pesquisa, quais são as soluções existentes para lidar com Requisitos de Qualidade negligenciados em Engenharia de Requisitos Ágeis em geral (não apenas em Larga Escala), conforme a literatura de Engenharia de Requisitos. Foram encontradas várias propostas de soluções que, no entanto, são em seu estágio inicial e, por sua vez, não são validados empiricamente. Isto concorda com a conclusão de (Inayat et al. apud ALSAQAF; DANEVA; WIERINGA, 2017) que observou falta de conhecimento sobre a efetividade das propostas de solução.

Algumas das soluções apresentadas introduziram vários novos artefatos para métodos ágeis, abordando específicos Requisitos de Qualidade. No entanto, pode-se pensar que a introdução de novos artefatos poderia resultar em tornar os métodos ágeis uma abordagem de documentação pesada que está desalinhada com o segundo valor do manifesto ágil, ou seja, “Trabalhando software sobre documentação abrangente”. Se uma solução para cada tipo de Requisito de Qualidade é proposta e adotada, acabamos com um grande número de abordagens. Ampliação ágil com ferramentas, tipos e / ou práticas necessárias para integrar cada tipo de Requisito de Qualidade vai certamente resultar em uma abordagem pesada que não atende aos princípios ágeis definidos no manifesto ágil.

## 2.5 Considerações Finais

Nesta seção podemos ter uma base de conhecimento sobre os principais conceitos de engenharia de requisitos, requisitos de qualidade e metodologias ágeis, como também uma breve explanação sobre os dois processos ágeis mais utilizados no mercado e a relação entre os dois assuntos na vivência da indústria de desenvolvimento de software. No ponto que

abordou os resultados da revisão sistemática da literatura, podemos perceber que várias publicações relataram a atenção insuficiente aos requisitos de qualidade em métodos ágeis, e foram encontradas várias propostas de soluções que, no entanto, estão em seu estágio inicial, necessitando mais estudos sobre estes assuntos.

### 3 Metodologia

Neste capítulo, serão expostos quais procedimentos foram adotados para atingir os objetivos estabelecidos para esta investigação. Desta forma, serão explanadas as técnicas, o esboço da amostra adotada para o estudo, como também a definição do tipo de abordagem, natureza e etapas da pesquisa.

#### 3.1 Classificação de pesquisa

A pesquisa é desenvolvida mediante o concurso dos conhecimentos disponíveis e a utilização cuidadosa de métodos, técnicas e outros procedimentos científicos. Na realidade, a pesquisa desenvolve-se ao longo de um processo que envolve inúmeras fases, desde a adequada formulação do problema até a satisfatória apresentação dos resultados (GIL, 2002).

Uma pesquisa pode ser caracterizada por diversos pontos de vista. Com relação a sua natureza, a presente investigação caracteriza-se por ser uma **pesquisa aplicada**. A mesma tem como objetivo gerar conhecimento para uma aplicação prática, visando solucionar problemas específicos existentes no cotidiano (PRODANOV, 2013). A pesquisa em questão tem o propósito de encontrar os problemas e desafios enfrentados pela indústria de desenvolvimento de software no que diz respeito ao uso de requisitos de qualidade em projetos ágeis.

De acordo com GIL (2002, p.41), “É sabido que toda e qualquer classificação se faz mediante algum critério. Com relação às pesquisas, é usual a classificação com base em seus objetivos gerais.” Com isso, esta pesquisa está classificada em três grandes grupos: exploratórias, descritivas e explicativas.

A presente análise é classificada como uma **pesquisa descritiva**, pois o seu objetivo primordial é a descrição das características de determinada população ou fenômeno ou, então, o estabelecimento de relações entre variáveis.

No momento em que se planeja uma pesquisa, deve-se selecionar um ambiente onde serão coletados os dados necessários para a investigação, além das técnicas de controle das variáveis envolvidas na pesquisa. Todos esses fatores estão combinados no delineamento da pesquisa. De acordo com GIL (2002, p.42), “ Uma de suas características mais significativas está na utilização de técnicas padronizadas de coleta de dados, tais como o questionário e a observação”. Neste caso foi adotada a técnica de **levantamento de campo ou Survey**.

Com relação ao modo como será abordado o problema analisado nesta investigação, foi adotado a **abordagem quantitativa** para traduzir em números as informações que poderão ser quantificadas, e posteriormente, analisadas e classificadas. Também foi adotada a **abordagem qualitativa**, para uma eventual necessidade de análise indutiva dos dados descritivos coletados no survey (PRODANOV, 2013).

### 3.2 O questionário

Segundo GIL (p.114, 2002), o questionário entende-se como um conjunto de questões que são respondidas por escrito pelo pesquisado. Essa técnica mostra-se bastante útil para a obtenção de informações acerca do que a pessoa "sabe, crê ou espera, sente ou deseja, pretende fazer, faz ou fez", bem como a respeito de suas explicações ou razões para quaisquer das coisas precedentes.

Com relação ao questionário da investigação em questão, este foi desenvolvido para retirar informações de um determinado grupo, mais especificamente os profissionais da área de tecnologia da informação. O objetivo foi reunir informações desses indivíduos no que diz respeito às suas experiências e opiniões sobre assuntos do âmbito da engenharia de software.

Vale salientar que o questionário desenvolvido para esta pesquisa buscou extrair de forma resumida, informações de múltiplos contextos do entrevistado. Como por exemplo, a sua origem, nível acadêmico e de experiência, o cargo que ocupa na empresa, o tamanho da organização na qual trabalha, quantos anos possui de experiência com metodologias ágeis, além da frequência com a qual o indivíduo já trabalhou com metodologias ágeis e a sua opinião sobre este paradigma combinado com as técnicas utilizadas para levantar requisitos junto com a vivência do uso de requisitos de qualidade em projetos ágeis.

Além das opiniões do entrevistado, o questionário buscou expor situações em que o cidadão questionado pudesse julgar determinados fatores que foram postos em questão para a sua análise. Desta forma, o questionário proverá um conjunto de dados quantificáveis para a investigação.

O questionário foi adaptado para ser compartilhado em um formato de formulário, viabilizado pela plataforma Google Forms. O período da coleta de dados foi a partir do dia 07 de novembro de 2018 até o dia 25 de novembro de 2018, para a sua divulgação, foram publicadas postagens em redes sociais, houve uma estratégia de divulgação para e-mails corporativos de empresas do segmento de TI localizadas em Recife, além do compartilhamento do link do formulário para diversas listas de e-mails acadêmicos, listas de algumas conferências especializadas no assunto e redes sociais. No fim, trinta e uma pessoas responderam o questionário. Ressalta-se ainda que o questionário completo está disponível no Apêndice I, no final deste documento.

### 3.3 Etapas de pesquisa

Esta pesquisa tem como objetivo fazer um estudo comparativo entre os achados da indústria e da literatura em relação ao uso de requisitos de qualidade em projetos ágeis.

Sendo assim, o estudo pode ser caracterizado como uma pesquisa de caráter exploratório, já que tem como objetivo tornar o assunto estudado mais familiar, deixá-lo mais claro, e gerar hipóteses (GIL, 2002).

Para que esta investigação pudesse obedecer critérios como coerência, consistência, originalidade e objetivação, foi necessário que todo o estudo fosse dividido em etapas:

- **Etapa 1 - Revisão bibliográfica:** Nesta fase, foi realizado todo o estudo que concerne à requisitos de qualidade, engenharia de requisitos e metodologias ágeis. Além disso, foi feita a leitura e análise da revisão sistemática da literatura que foi mencionada na seção 2 que explana sobre a problemática dos desafios enfrentados pelo uso de requisitos de qualidade em projetos ágeis.
- **Etapa 2 - Construção do questionário:** Após obter todo um apanhado teórico necessário para entender e discutir sobre a problemática em questão, esta etapa serviu para elaborar um questionário que gerará insumos para esta pesquisa.
- **Etapa 3 - Análise e cruzamento dos dados:** Com os dados oriundos do questionário desenvolvido, esta etapa teve a função de analisar as respostas dos entrevistados. Com o conhecimento gerado pela análise, ocorreu um cruzamento de informações gerados do questionário e dos resultados obtidos pela revisão sistemática da literatura que serviu como referência para este trabalho. É bom ressaltar que das três questões de pesquisas que contém no artigo analisado, foram abordadas duas.
  - A terceira questão de pesquisa questionava quais são as soluções existentes para lidar com Requisitos de Qualidade negligenciados em Engenharia de Requisitos ágil em geral (não apenas em ALSD), conforme a literatura RE e o motivo de não analisar esta questão é que a maioria das soluções propostas são criações de novos artefatos, sem contar que estas não foram consideradas como propostas ágeis, de acordo com o grau de agilidade que foi avaliado com base na presença de cinco elementos-chave: flexibilidade, velocidade, simplicidade, aprendizagem e capacidade de resposta.
  - E de acordo com os autores, se uma solução para cada tipo de QRs é proposta e adotada, acabamos com um grande número de abordagens. Ampliação ágil com ferramentas, tipos e/ou práticas necessárias para integrar cada tipo de QRs vai certamente resultar em uma abordagem pesada que não atende aos princípios ágeis definidos no manifesto ágil.
- **Etapa 4 - Conclusão dos resultados:** Por fim, nesta fase foi realizada a interpretação e conclusão dos resultados oriundos do cruzamento das informações presentes na etapa anterior.

### 3.4 Considerações finais

Esta seção expôs a importância de como uma pesquisa científica deve seguir um procedimento, para que desta forma, a investigação obtenha uma confiabilidade e esteja de acordo com as exigências científicas necessárias para a sua validade.

## 4. Análise dos resultados

Nesta seção, serão explanados os resultados provenientes do questionário que foi compartilhado junto aos profissionais do mercado de TI. Desta forma, será ilustrada a

realidade, através da opinião do público entrevistado, do uso dos requisitos de qualidade no contexto ágil.

#### 4.1 Características gerais da amostra

O Survey elaborado para gerar insumos para a presente pesquisa foi compartilhado por diversos canais de comunicação para diversas partes do país, para que conseguíssemos extrair opiniões de diferentes contextos. Após a conclusão do compartilhamento do questionário, foram obtidas 31 respostas de entrevistados de diversos estados, além de uma variedade de graus de formação e tempo de experiência na área de TI. As respostas serão explanadas logo adiante.

De acordo com os resultados no Gráfico 1, grande parte dos entrevistados é oriundo do estado de Pernambuco, parcela essa que resultou 41,9% dos entrevistados, seguido do estado da Paraíba com uma fatia de 29% dos participantes. A região Nordeste obteve o maior número de respostas, totalizando 77,3% das respostas, que também contou com o estado do Ceará, representando 3,2% do total de respostas e o estado de Alagoas também com 3,2% dos participantes. Já a região Sul, contou com a participação do estado do Rio Grande do Sul, parcela essa que resultou 12,9% dos entrevistados, seguido do estado de Santa Catarina, com uma fatia de 3,2% das respostas obtidas. A região sudeste foi representada apenas pelo estado de São Paulo, com um resultado de 6,5% dos entrevistados.

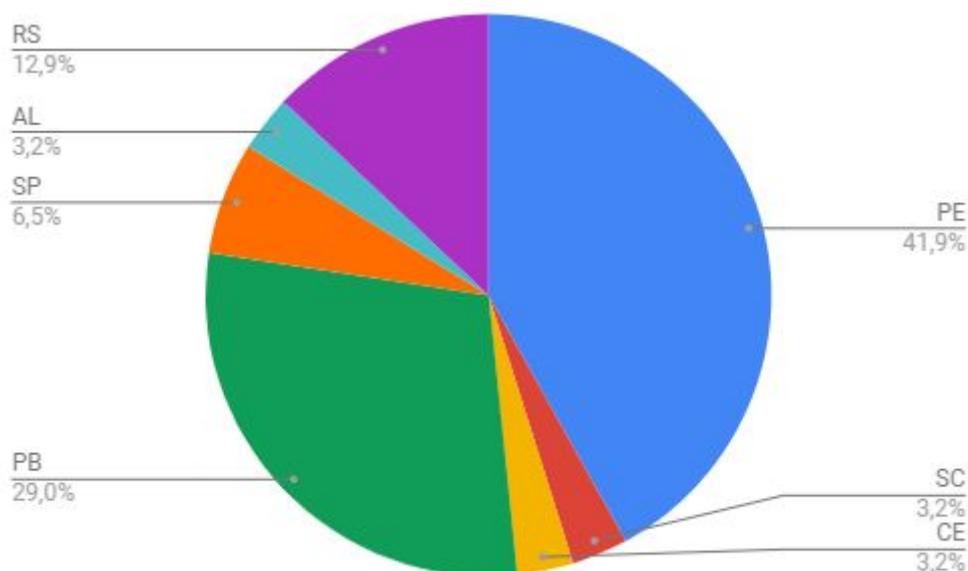


Gráfico 1: Classificação de entrevistados conforme à área geográfica  
(Fonte: O Autor)

Com relação ao grau de formação dos profissionais entrevistados, o Gráfico 2 mostra que 38,7%, possuem curso superior completo, um pouco mais do percentual de pessoas que

possuem curso superior incompleto que foi de 32,3%. Em contrapartida, 19,4% dos entrevistados possuem mestrado e 9,7% do total de entrevistados possuem doutorado.

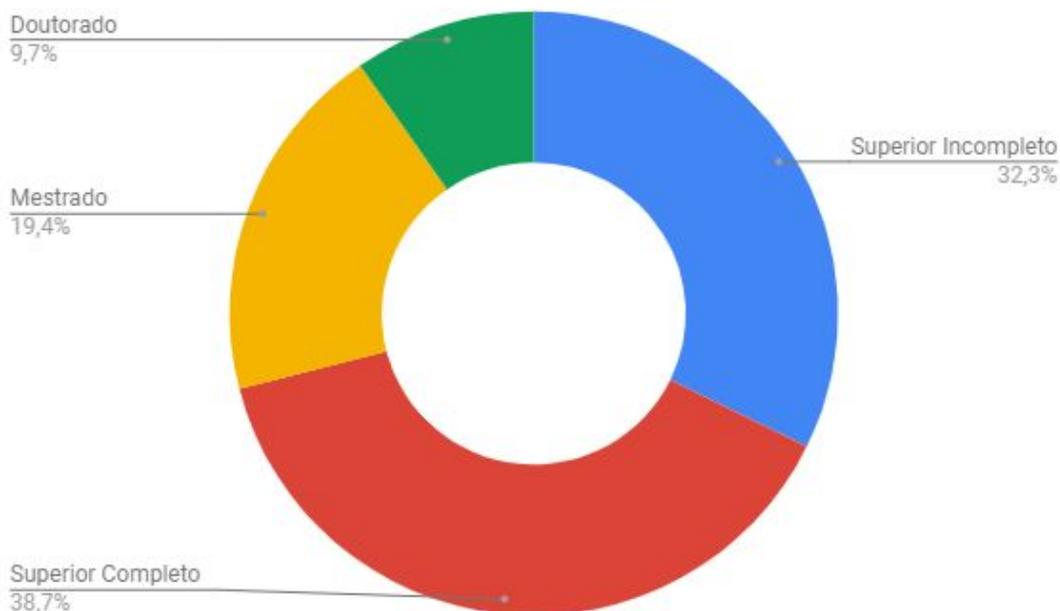


Gráfico 2: Classificação dos entrevistados conforme o seu grau de formação  
(Fonte: O Autor)

Em relação à variável tempo de experiência, 29% dos entrevistados possuem experiência no intervalo entre 3 a 5 anos, seguidos por 25,8% de profissionais que possuem experiência dentro de um intervalo entre 0 e 3 anos, mesmo percentual de pessoas que possuem prática em um intervalo entre 5 a 10 anos. Por fim, 16,1% dos entrevistados possuem vivência de mais de 15 anos no mercado de TI e apenas 3,2% dos entrevistados possuem entre 10 e 15 anos de experiência.

A pesquisa também mostra os cargos dos entrevistados atualmente, no Gráfico 3, constatou-se que cerca de 15 respondentes, o que representa 48,4% do total de respostas, ocupam o cargo de desenvolvedores de sistemas. 12,9% deles são gerentes de projetos (Project Manager), 9,7% deles são da área de qualidade/testes de software, 6,5% são analistas de negócios, o mesmo percentual de 6,5% também são donos dos produtos (Product Owner), já os cargos de analista de suporte, devops, analista de sistema, professor e gerente técnico de arquiteto representam 3,2% cada.

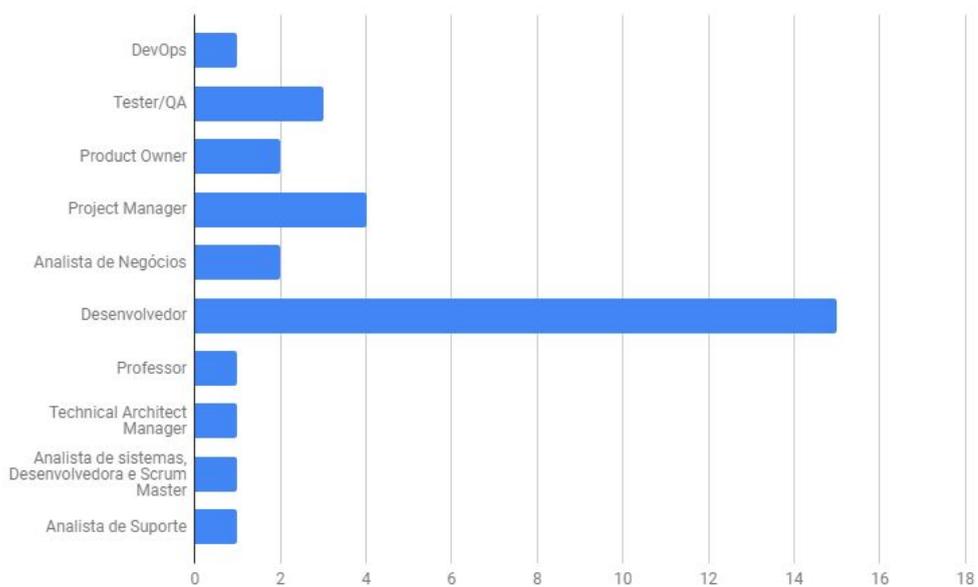


Gráfico 3: Classificação dos entrevistados conforme o seu cargo atual  
(Fonte: O Autor)

A presente pesquisa também buscou saber qual o porte da empresa na qual os seus entrevistados estão desempenhando suas funções, conforme exposto no Gráfico 4. Diante disso, constatou-se que 48,4% dos entrevistados trabalham em empresas de grande porte, seguido por 29% dos entrevistados trabalhando em pequenas empresas, como também uma parcela de 12,9% de profissionais que trabalham em empresas de médio porte, e por fim, uma fatia de 9,7% de pessoas questionadas que trabalham em microempresas.

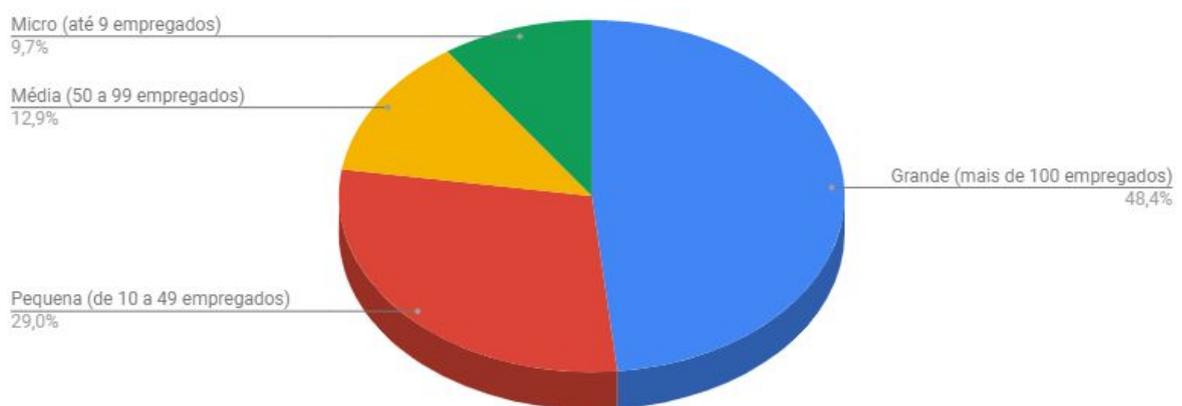


Gráfico 4: Percentual dos entrevistados conforme o porte da empresa na qual trabalham  
(Fonte: O Autor)

A pesquisa também procurou descobrir quantos anos de experiência os entrevistados tinham com metodologias ágeis. Foi constatado que cerca de 15 respondentes, o que representa 48,4% do total de respostas, possuem experiência de 1 a 3 anos. 32,3% deles têm de 3 a 5 anos, 9,7% deles têm de 5 a 10 anos, 6,5% não tem nenhuma experiência e por fim 3,2% tem experiência de 10 a 15 anos.

Em relação à frequência com que os profissionais da área de TI utilizam metodologias ágeis para o desenvolvimento de software. Notou-se que 54,8% dos profissionais utilizam métodos ágeis em uma média de 2 a 5 projetos ao ano. Por outro lado, 19,4% dos entrevistados utilizam processos ágeis em 6 ou mais projetos ao ano, mesmo percentual de profissionais que utilizam esse paradigma em apenas 1 projeto ao ano. Por fim, apenas 6,5% dos entrevistados afirmaram que não utilizam metodologias ágeis no dia-a-dia dos seus projetos. Com isso, tais percentuais comprovam a disseminação e a crescente utilização das técnicas ágeis no mercado de TI, para que desta forma a área de desenvolvimento de software consiga acompanhar as tendências e competitividade do mercado atual.

Com relação ao levantamento de requisitos, buscamos através da presente pesquisa, analisar quais técnicas estão sendo as mais utilizadas em conjunto com as metodologias ágeis. No Gráfico 5 reflete a realidade encontrada para este contexto no mercado atual.

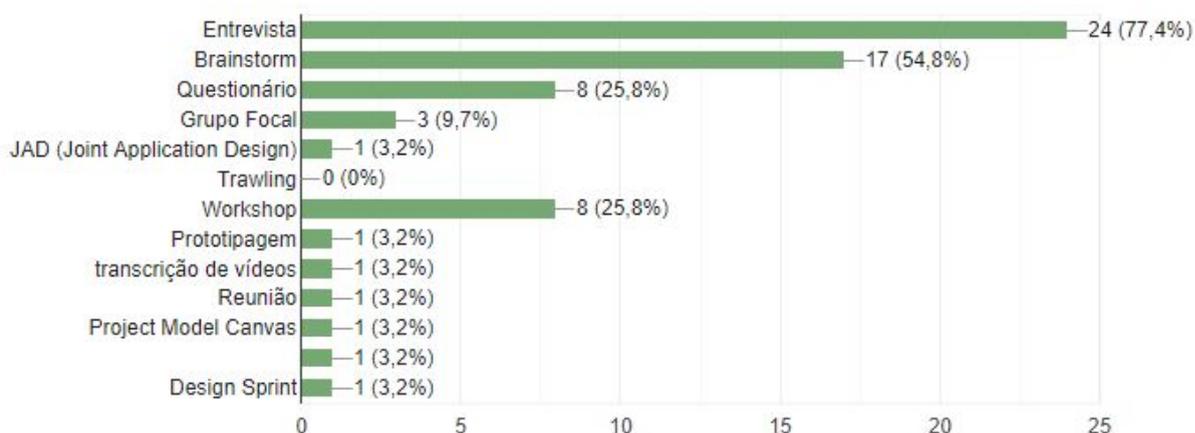


Gráfico 5: Percentual das técnicas utilizadas para levantamento de requisitos  
(Fonte: O Autor)

De acordo com os dados obtidos, as técnicas mais utilizadas para o levantamento de requisitos são a Entrevista, com 77,4% das respostas, e o Brainstorm, conforme 54,8% dos entrevistados. Outras duas técnicas bastante utilizadas são o Questionário e o Workshop, ambas com 25,8% de preferência dos profissionais.

A presente pesquisa também buscou coletar, através da pergunta de número dez<sup>1</sup>, os aspectos positivos das técnicas de levantamento de requisitos.

<sup>1</sup> Pergunta 10: De acordo com os métodos que você assinalou acima, cite os pontos positivos da utilização de cada um deles.

Diante da maioria das respostas dos entrevistados, a entrevista tem o objetivo de propor um melhor entendimento do negócio, para que desta forma a equipe técnica possua insumos suficientes para o desenvolvimento do plano do projeto. Esta técnica proporciona ainda uma compreensão da realidade dos usuários, colhendo requisitos de forma rápida e barata. Além de ajudar a entender melhor a necessidade do cliente e a ter mais empatia com o contexto analisado.

Segundo a resposta do entrevistado P25 para a pergunta de número 10, “a entrevista com usuários permite uma visão totalmente operacional dos requisitos”. Além disso, o participante P26 informa que os benefícios são: “Colher informações mais precisas do cliente. Já que o mesmo terá mais tempo para falar. Além disso o entrevistador pode fazer perguntas que não estavam planejadas”. Já para o entrevistado P05 ele informa que: “É mais fácil entender a necessidade do cliente e sugerir alternativas mais viáveis”. No geral, os entrevistados caracterizam a entrevista como uma técnica ágil, direta e informal, viabilizando que a equipe técnica obtenha conhecimento de forma rápida e tenha a possibilidade de abrir uma discussão com o cliente sobre as questões encontradas, e um melhor entendimento do problema do cliente, gerando assim, um bom relacionamento com o mesmo.

Já o Brainstorm, também foi relatado nas respostas provenientes da pergunta dez. De acordo com os entrevistados, essa técnica favorece uma maior gama de opções com relação a pontos específicos do software, pois é uma técnica rápida e eficiente. Ele é uma boa fonte de ideias, se for bem gerenciado, pois muitas ideias interessantes são disseminadas com esta técnica.

Alguns relatos dos participantes explicita alguns pontos positivos sobre esta técnica, como por exemplo: “É possível obter o feedback direto e imediato do cliente”. (P31); “O respaldo de ideias de demais colaboradores de outras áreas de atuação”. (P24); “Contato com a equipe, diversidade de ideias e a linha do que o cliente deseja, assim sendo assertivo na execução”. (P14).

Com isso, pode-se obter um consenso entre a diversidade de informações e possíveis soluções. Porém, deve-se ter um controle e foco atrelado ao seu uso, para que não ocorra um caos de ideias e tempo perdido no trabalho de possíveis soluções que não serão viáveis para o software.

Já o questionário, de acordo com as opiniões referentes à pergunta dez, pode ser utilizado para entender problemas específicos do cliente com mais rapidez, possibilitando obter uma maior abrangência do que é necessário saber. Segundo a resposta do entrevistado P02 para a pergunta de número 10, “ajuda a obter resultados mais rápidos e com mais praticidade.” No entanto, ao fazer o planejamento do questionário, deve ficar atento à quem vai responder, para que as perguntas sejam construídas de acordo com os objetivos específicos de cada um.

A pesquisa também buscou coletar, através da pergunta de número onze<sup>2</sup>, os aspectos negativos das técnicas de levantamento de requisitos.

---

<sup>2</sup> Pergunta 11: De acordo com os métodos que você assinalou acima, cite os pontos negativos da utilização de cada um deles.

Os profissionais entrevistados apontaram, nas respostas da pergunta onze alguns pontos negativos relacionados à esta técnica. Como por exemplo: “Às vezes o cliente perde o foco e acaba não discutindo tudo o que gostaria. Pode ser demorado. Muitas vezes o cliente esquece o que disse ou muda de opinião” (P05);

“Disponer de tempo para a sua realização, para elaborar um roteiro coerente, ao invés de entrevistar aleatoriamente.” (P02);

Em alguns casos, o consumidor não tem uma noção clara da sua necessidade. Além disso, se o entrevistador não possuir um bom roteiro e conhecimento, serão necessárias várias sessões de entrevistas para mapear tudo o que se necessita.

Com isso, observa-se que para possuir bons resultados mediante a entrevistas, é necessário tempo, conhecimento e planejamento. Porém, em casos específicos, ainda não se consegue obter a real necessidade do cliente, pois o mesmo muitas vezes pode não demonstrar tudo aquilo que deseja, acarretando um retrabalho no futuro.

Já o questionário, de acordo com as opiniões referentes à pergunta onze foi visto alguns relatos como: “Às vezes a conclusão dos resultados de um questionário pode ficar meio vaga, principalmente se ele tiver muitas perguntas abertas, sendo necessário elaborar outro mais específico e que no final acaba se tornando mais trabalhoso”. (P02);

“Dependendo da forma como é feito, ele pode enviesar as respostas, não dando espaço para o surgimento de outras ideias”. (P23). Com isso, observa-se que para possuir bons resultados mediante a esta técnica, é necessário que seja muito bem elaborado e atentar a quem vai respondê-lo, para que a elaboração do mesmo seja baseado no público alvo específico.

Com isso, entende-se que cada técnica possui suas particularidades e deve ser utilizada de acordo com a necessidade específica de cada um, além de que os pontos negativos apontados pelos entrevistados devem ser notados pelos utilizadores das técnicas para não haver repetição dos problemas já indicados.

Entrando no foco principal da pesquisa que é relacionado ao uso de requisitos de qualidade em projetos ágeis, o questionário da pesquisa em questão buscou colher a opinião dos entrevistados com relação à vivência em 12 desafios específicos causados por práticas ágeis em relação aos Requisitos de Qualidade numa escala de zero à cinco, de responsabilidade que cada desafio específico em projetos ágeis analisado, contribui para a negligência dos Requisitos de Qualidade em projetos ágeis.

Essa escala de zero à cinco é relacionado a vivência dos entrevistados de cada desafio específico onde:

- 0 significa que a afirmação nunca foi vivenciada por vocês;
- 1 representa que você discorda completamente da afirmação;
- 2 representa que você discorda parcialmente da afirmação;
- 3 representa que você não tem certeza se discorda ou concorda com a afirmação;
- 4 representa que você concorda parcialmente com a afirmação;
- 5 representa que você concorda completamente com a afirmação.

Com isso, notou-se que com relação ao desafio “Métodos ágeis não fornecem uma técnica amplamente aceita para reunir os Requisitos de Qualidade”. Numa escala de 0 a 5, o grau que teve mais votos foi o número 2, conforme exposto no Gráfico 6. Constatou-se que

29% dos entrevistados discordam parcialmente da afirmação; Seguido por 19,4% dos entrevistados que não têm certeza se discordam ou concordam com a afirmação; Como também uma parcela de 16,1% dos participantes discordam completamente da afirmação; Em seguida com a mesma porcentagem de 12,9% os participantes concordam parcialmente e completamente com a afirmação; Por fim, uma fatia de 9,7% de pessoas questionadas nunca vivenciaram esse desafio.

(a) Métodos ágeis não fornecem uma técnica amplamente aceita para reunir os Requisitos de Qualidade.

31 respostas

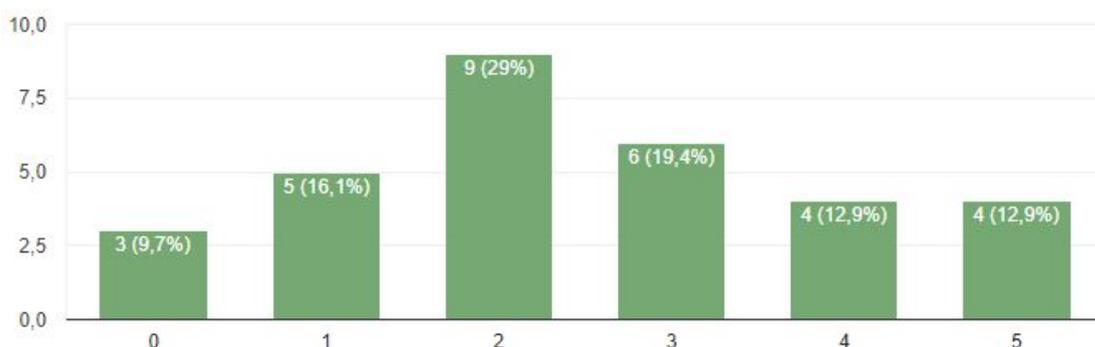


Gráfico 6: Percentual dos entrevistados conforme o desafio específico 1  
(Fonte: O Autor)

Por outro lado, quatro dos 12 desafios explanados estão relacionados ao papel do product owner, que são eles:

1. “Dependência do proprietário/representante do produto (Product Owner) como ponto único para coletar os requisitos”, o qual teve como resultado 11 votos, que representa 35,5% do total dos entrevistados no número 4 que representa que os participantes concordam parcialmente com a afirmação; Para mais informações detalhadas sobre esse desafio, verificar no Gráfico 7.
2. “Disponibilidade insuficiente do proprietário/representante do produto (Product Owner)”, o qual teve como resultado 29% do total de participantes, que representa 9 votos, no número 4, onde os participantes concordam parcialmente com a afirmação; Para mais informações detalhadas sobre esse desafio, verificar no Gráfico 8.
3. “Falta de conhecimento específico do proprietário/representante do produto (Product Owner)” obteve como resultado 12 votos do total de entrevistados, representando 38,7%, no número 4, onde os participantes concordam parcialmente com a afirmação. Para mais informações detalhadas sobre esse desafio, verificar no Gráfico 9;
4. “Sobrecarga de trabalho do proprietário/representante do produto (Product Owner)” obteve como resultado 38,7%, representando 12 votos dos participantes, no número 3,

onde eles não têm certeza se discordam ou concordam com a afirmação. Para mais informações detalhadas sobre esse desafio, verificar no Gráfico 10.

Para se ter uma visão geral dos resultados detalhados dos desafios relacionados ao papel do product owner, os Gráficos 7, 8, 9, 10 serão apresentados a seguir.

**(b) Dependência do proprietário/representante do produto (Product Owner) como ponto único para coletar os requisitos.**

31 respostas

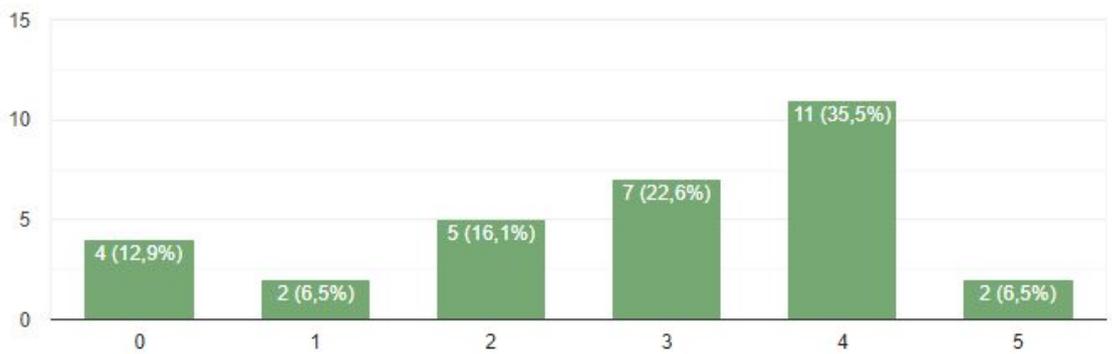


Gráfico 7: Percentual dos entrevistados conforme o desafio específico 2  
(Fonte: O Autor)

**(d) Disponibilidade insuficiente do proprietário/representante do produto (Product Owner).**

31 respostas

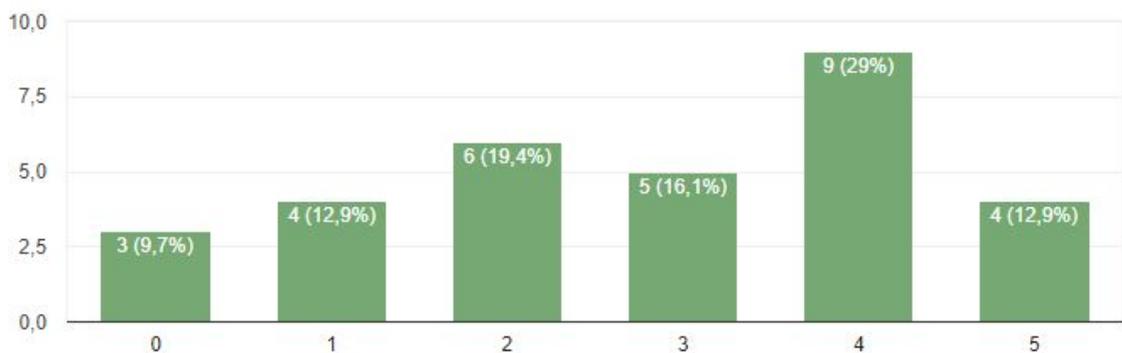


Gráfico 8: Percentual dos entrevistados conforme o desafio específico 4  
(Fonte: O Autor)

(g) Falta de conhecimento específico do proprietário/representante do produto (Product Owner).

31 respostas

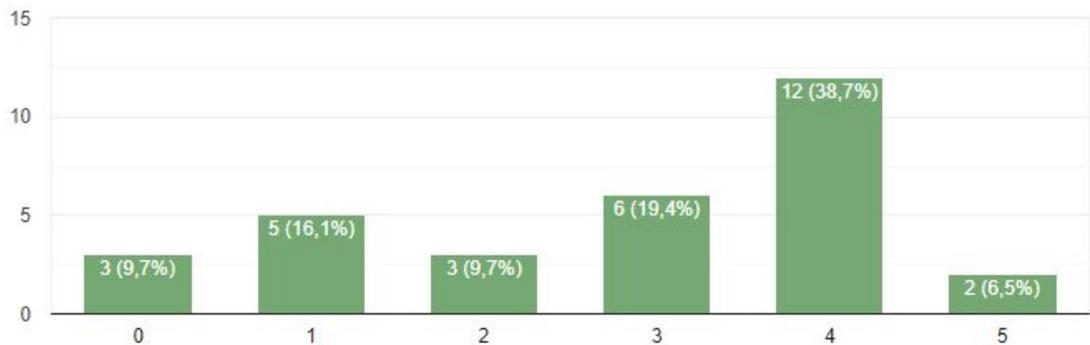


Gráfico 9: Percentual dos entrevistados conforme o desafio específico 7  
(Fonte: O Autor)

(i) Sobrecarga de trabalho do proprietário/representante do produto (Product Owner).

31 respostas

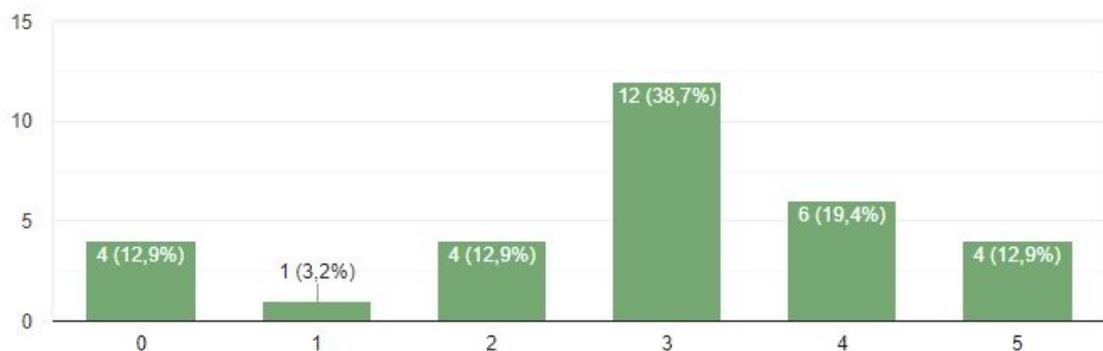


Gráfico 10: Percentual dos entrevistados conforme o desafio específico 9  
(Fonte: O Autor)

Em relação ao desafio "Incapacidade de histórias de usuários para documentar os Requisitos de Qualidade", este obteve como maior resultado o número 1, representando 25,8% dos votos, onde os participantes discordam completamente da afirmação. Sobre o desafio "A validação dos Requisitos de Qualidade ocorre tarde demais no processo", este teve como maior resultado o número 4, representado por 7 votos ou 22,6% do total de entrevistados que concordam parcialmente com a afirmação, mais informações detalhadas podem ser vistas no Gráfico 11.

Já o desafio "Falta de mecanismos de rastreabilidade de requisitos" obteve como maior resultado o número 4, representando 25,8% dos votos, onde os participantes concordam parcialmente com a afirmação, verificar no Gráfico 12.

Em relação ao desafio "Análise insuficiente dos requisitos", este teve como maior resultado o número 4, representado por 35,5% do total de participantes que concordam parcialmente com a afirmação, para mais informações detalhadas, verificar no Gráfico 13.

Já o desafio "Ignorar os requisitos de arquitetura previsíveis" obteve como maior resultado o número 3, representando 25,8% dos votos, onde os participantes não tem certeza se discordam ou concordam com a afirmação.

Sobre o desafio "Incapacidade de histórias do usuário para documentar dependências de requisitos", este teve como maior resultado o número 3, representado por 9 votos ou 29% do total de entrevistados que não têm certeza se discordam ou concordam com a afirmação, verificar no Gráfico 14.

Por último, o desafio "Foco no fornecimento de funcionalidade ao custo da flexibilidade da arquitetura" obteve como maior resultado o número 3, representando 38,7% dos votos, onde os participantes não têm certeza se discordam ou concordam com a afirmação.

Para se ter uma visão geral dos resultados detalhados dos desafios específicos, os Gráficos 11, 12, 13, 14 serão apresentados a seguir.

### (e) A validação dos Requisitos de Qualidade ocorre tarde demais no processo.

31 respostas



Gráfico 11: Percentual dos entrevistados conforme o desafio específico 5

(Fonte: O Autor)

(f) Falta de mecanismos de rastreabilidade de requisitos.

31 respostas

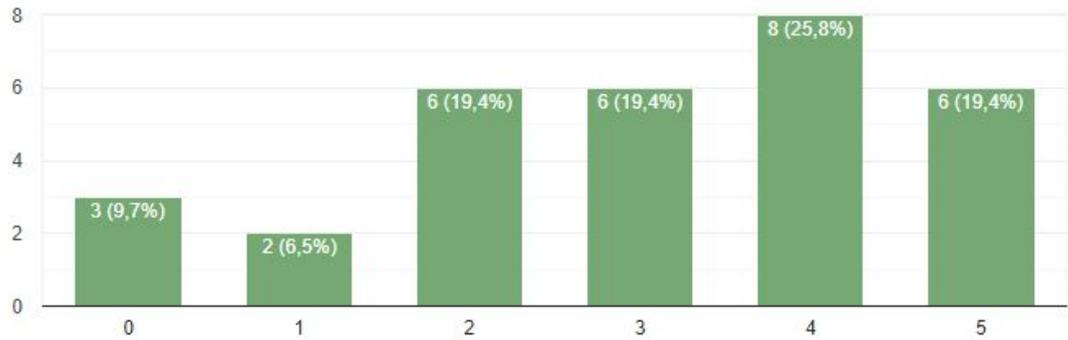


Gráfico 12: Percentual dos entrevistados conforme o desafio específico 6  
(Fonte: O Autor)

(h) Análise insuficiente dos requisitos.

31 respostas

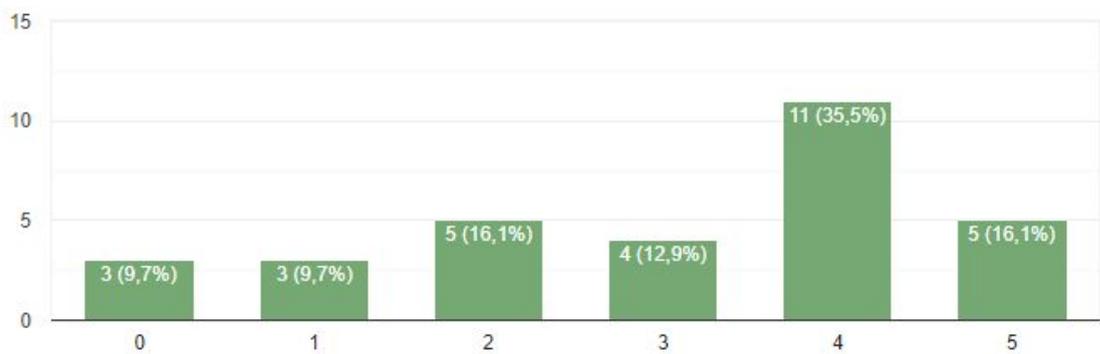


Gráfico 13: Percentual dos entrevistados conforme o desafio específico 8  
(Fonte: O Autor)

(k) Incapacidade de histórias do usuário para documentar dependências de requisitos.

31 respostas

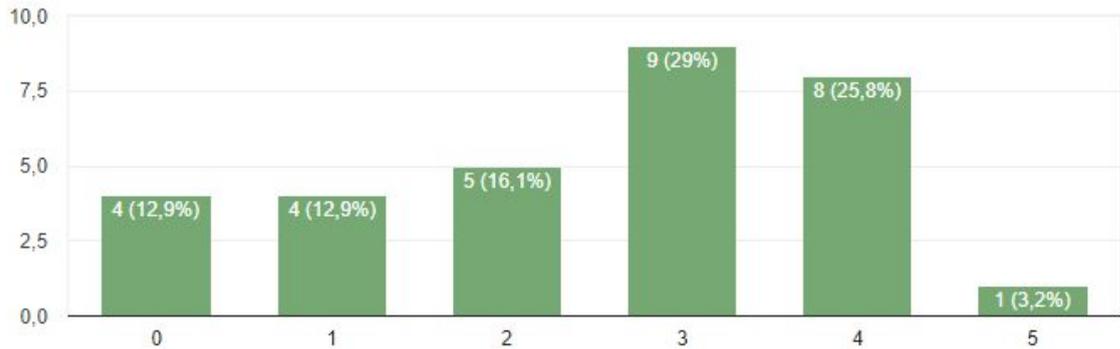


Gráfico 14: Percentual dos entrevistados conforme o desafio específico 11  
(Fonte: O Autor)

A presente pesquisa também buscou junto aos entrevistados saber as práticas de Engenharia de Requisitos mais vivenciadas pelos entrevistados em projetos de desenvolvimento ágil de software para lidar com os Requisitos de Qualidade. Conforme o Gráfico 15, os entrevistados relataram que as principais práticas vivenciadas são:

- “*Comunicação face a face*: Sessões de comunicação entre a equipe de desenvolvimento e o Dono/Representante do Produto (Product Owner) para identificar os requisitos”. Com 23 votos, resultando em 74%.
- “*Os requisitos emergem durante o processo de desenvolvimento em vez de serem definidos antecipadamente*”, obtendo 19 votos, resultando em 61,3%.
- “*Priorização frequente de requisitos*: Prioriza continuamente as histórias do usuário com base nas mudanças dos valores do negócio”, com 17 votos, resultando em 54,8%.
- “*Histórias de usuários*: Descrição breve e abstrata dos requisitos usados para incentivar a comunicação face a face”, também com 17 votos, resultando em 54,8%.

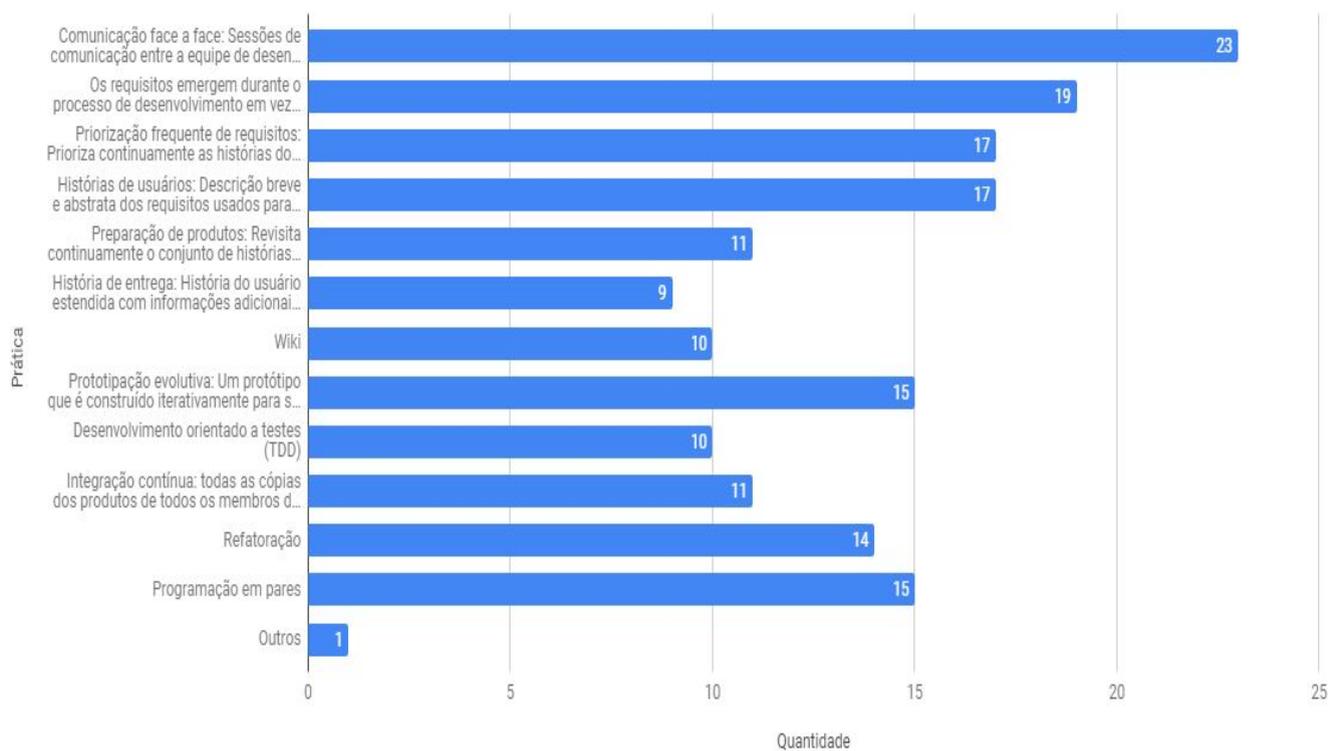


Gráfico 15: Percentual dos entrevistados conforme as práticas específicas  
(Fonte: O Autor)

Segundo as opiniões dos profissionais que responderam a pergunta anterior, podemos visualizar alguns relatos positivos e negativos em relação às práticas vivenciadas, como por exemplo: “Acredito que a descrição dessas práticas já diz o benefício de cada uma delas. Os pontos negativos no geral são as análises dos requisitos tomarem muito tempo do projeto devido à diversos fatores como falta de entendimento do requisito, a mudança de requisitos de última hora, a adição de um novo requisito, etc”. (P01).

Segundo o relato do participante P31, “A Prototipação evolutiva evita retrabalho. Além de atingir a fase de requisito, esta ação faz parte da fase de desenvolvimento; A Integração contínua facilita muito a gerência dos artefatos de software; Comunicação face a face evita falta de entendimento”.

Já o entrevistado P27 informou que “A comunicação face a face permite resolver pendências no momento em que surgem; Wiki é importante para não centralizar o conceito em uma pessoa, mas em um local onde todas as dúvidas foram solucionadas e transformadas em um documento que guia o desenvolvimento; A prototipação é importante para garantir que houve entendimento mútuo do que precisa ser feito. Integração contínua para rastrear a evolução da solução e suas versões. Refatoração para garantir a qualidade do código”.

O participante P24 relatou, “Positivo é não ficar engessado em documentação, e a parte negativa é que mudanças ocorrem a todo tempo, e não se tem ideia do que se quer (requisitos) até o projeto está sendo construído. Assim, as coisas podem mudar completamente no meio do desenvolvimento, gerando retrabalho para coisas simples como

campos obrigatórios, restrições, permissões, ou coisas que haviam sido consideradas que não teria e agora passa a ter”.

Também foi possível notar comentários interessantes sobre práticas que não ficaram entre as mais votadas, como a contribuição do entrevistado P15, “*TDD*: É uma ótima prática para desenvolvedores criarem hábito para focar no atendimento do assertivo requisito, mas acaba sendo uma prática que exige uma mudança de hábito para alguns desenvolvedores.

*Programação em pares*: Prática que até agora tenho pouca experiência, mas que permite troca de conhecimento, mesmo que minimamente, entre os desenvolvedores que estão trabalhando em pares no momento.

*Refatoração*: Até agora não vi ponto negativos, é um hábito que melhora muito a qualidade do código em si, o tornando mais legível para outros desenvolvedores, como um desenvolvedor que trabalha com códigos pós-refatoração, sinto que a qualidade do trabalho se eleva”.

Podemos notar que as práticas listadas no questionário tem um número relevante de vivências pelos entrevistados, pois das 12 práticas, apenas uma não teve o mínimo de 9 votos, ou 29% do total de participantes, que foi a *Desenvolvimento orientado a testes (TDD)*. Um ponto interessante que é bom destacar que a maioria dos entrevistados utilizam várias práticas em conjunto e não apenas uma ou duas específicas, fazendo com que tenha mais agilidade na geração de artefatos funcionais em cada iteração.

A presente pesquisa também buscou junto aos entrevistados saber o que mais causa problemas em projetos ágeis, diante de dois quesitos: cliente e processo. Com isso, diante do quesito cliente, foi identificado que os fatores que mais ocorrem problemas nos projetos é a interação inadequada entre cliente e equipe técnica, com um total de 21 votos, representado por 67,7%, seguido da pouca disponibilidade do cliente que teve 61,3% e em terceiro os usuários não sabem o que eles querem com 58,1% do total de votos dos entrevistados. Para uma melhor visualização destes dados, no Gráfico 16 mostra a recorrência de relatos dos usuários com relação a este ponto.

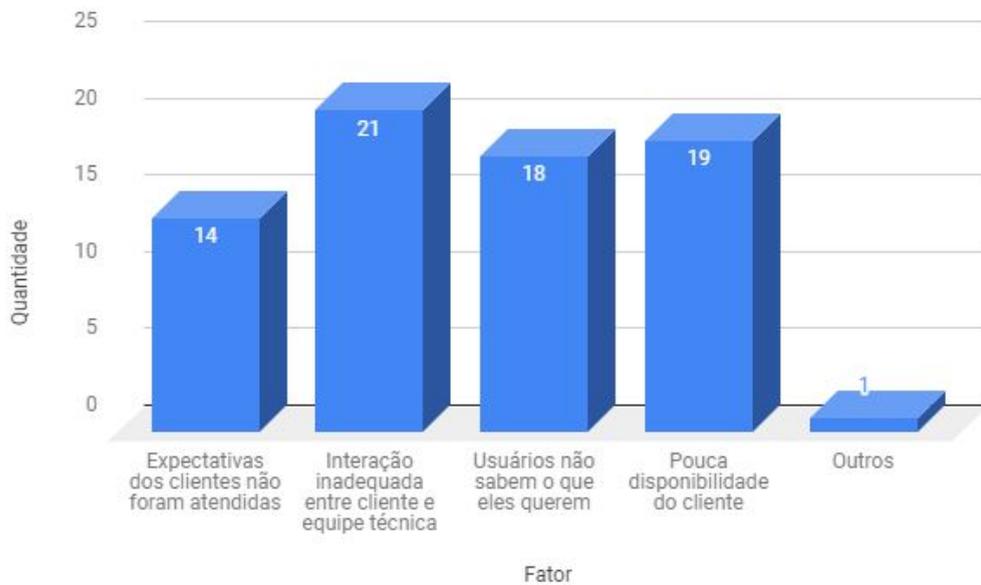


Gráfico 16: Recorrência das causas de problemas relacionados aos clientes em projetos ágeis: (Fonte: O Autor)

Já com relação aos fatores contribuintes para contratempos no contexto dos processos, as razões mais mencionadas pelos entrevistados foram a falta de perspectiva do cliente na fase de validação de requisitos juntamente com os requisitos não acordados com a equipe de desenvolvimento, ambos representados com 51,6%, ou 16 votos, do total de votos dos entrevistados. A ineficiência na análise e inspeção de requisitos também foi um dos fatores mais votados, obtendo 15 votos, ou 48,4% do total de votos. O Gráfico 17 apresenta as respostas mais detalhadas.

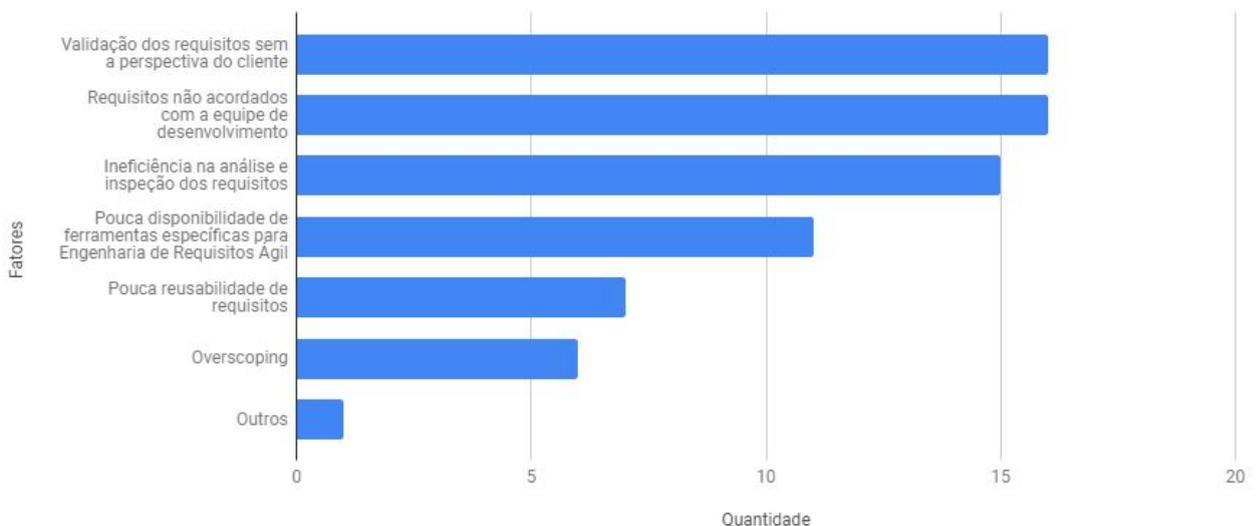


Gráfico 17: Esboço das causas referentes aos processos adotados nos projetos ágeis. (Fonte: O Autor)

## 4.2 Resultados referentes ao cruzamento das informações do survey e da revisão sistemática da literatura

Nesta seção serão expostos os resultados do cruzamento das informações provenientes da pesquisa em questão e da revisão sistemática da literatura que foi adotada como referência principal para este trabalho.

De acordo com a RQ1, que questionava quais práticas ágeis eram usadas para projetar Requisitos de Qualidade em ambientes de larga escala, de acordo com a literatura publicada, mesmo sem a intenção dos autores foram encontrados artigos que analisaram quatro tipos de Requisitos de Qualidade: Segurança, Safety, Conformidade e Usabilidade. Cada um desses artigos enfocou as práticas específicas atualmente usadas para projetar os respectivos tipos de Requisitos de Qualidade discutidos no trabalho, independentemente de outros Requisitos de Qualidade. Em essência, essas práticas específicas sugerem a introdução de um novo artefato em Engenharia de Requisitos Ágeis, ou um novo papel, ou novo comportamento social, ou uma nova ferramenta, ou uma combinação de todos. Um resumo dessas práticas pode ser encontrado na Tabela 2, onde a coluna mais à direita da tabela indica se a prática diz respeito a artefatos, ferramentas, papéis ou comportamento social.

O Survey procurou conferir a adesão dessas práticas explanadas pela revisão sistemática na indústria. Segundo as respostas do survey, foi constatado a grande aceitação e uso dessas práticas listadas na Tabela 2, obtendo um número relevante de vivências pelos entrevistados. Das 12 práticas, apenas uma não teve o mínimo de 9 votos, ou 29% do total de participantes, que foi a *Desenvolvimento orientado a testes (TDD)*. Um ponto interessante que é bom destacar é que a maioria dos entrevistados utilizam várias práticas em conjunto e não apenas uma ou duas específicas, fazendo com que tenha mais agilidade na geração de artefatos funcionais em cada iteração, mais detalhes sobre esses dados podem ser vistas no Gráfico 15.

Com relação à RQ2, os pesquisadores buscaram saber os desafios de Requisitos de Qualidade que foram relatados em projetos ágeis, em geral. Vinte e seis artigos relataram explicitamente e implicitamente a negligência dos Requisitos de Qualidade em projetos ágeis. A Tabela 3 resume 12 desafios específicos enfrentados por práticas ágeis em relação aos Requisitos de Qualidade em geral, de acordo com esses artigos. A Tabela 3 sugere que algumas características de Engenharia Ágil de Requisitos são mais fortes em livros didáticos do que na vida real (por exemplo, o papel do product owner e o uso de histórias de usuários), onde pode ser considerado de fato como inibidor na engenharia de requisitos de qualidade. Quatro dos 12 desafios estão relacionados ao papel do product owner. A este respeito, alguns artigos apontam para vários aspectos associados a esse papel como condutores de dificuldades de Requisitos de Qualidades, como por exemplo:

1. Falta de conhecimento específico do proprietário/representante do produto (Product Owner);
2. Sobrecarga de trabalho do proprietário/representante do produto (Product Owner);
3. Dependência do proprietário/representante do produto (Product Owner) como ponto único para coletar os requisitos;

4. Disponibilidade insuficiente do proprietário/representante do produto (Product Owner).

Isso mostra que o Product Owner é uma fonte potencial de desafios de Engenharia de Requisitos.

Com isso o Survey buscou colher a opinião dos entrevistados com relação à vivência nesses 12 desafios específicos enfrentados por práticas ágeis em relação aos Requisitos de Qualidade. Numa escala de zero à cinco o respondente deveria indicar o grau de responsabilidade em que cada desafio específico analisado contribui para a negligência dos Requisitos de Qualidade em projetos ágeis. Essa escala de zero à cinco é relacionado a vivência dos entrevistados de cada desafio, onde 0 significa que a afirmação nunca foi vivenciada pelo entrevistado; 1 representa que ele discorda completamente da afirmação; 2 representa que ele discorda parcialmente da afirmação; 3 representa que ele não tem certeza se discorda ou concorda com a afirmação; 4 representa que ele concorda parcialmente com a afirmação; 5 representa que ele concorda completamente com a afirmação.

Segundo as respostas do survey, foi constatado que o desafio específico “Incapacidade de histórias do usuário para documentar dependências de requisitos” obteve 29% ou 9 votos como maior resultado o grau 3, mostrando que os entrevistados não têm certeza se discorda ou concorda com a afirmação. Logo em seguida, com 25,8% ou 8 votos, os participantes escolheram a opção 4 que representa que eles concordam parcialmente com a afirmação. Isso mostra um número alto de concordância entre os resultados da RSL e os resultados dos entrevistados.

Em relação ao papel do product owner, o survey destacou quatro desafios específicos, que foram eles:

- “Dependência do proprietário/representante do produto (Product Owner) como ponto único para coletar os requisitos”;
- “Disponibilidade insuficiente do proprietário/representante do produto (Product Owner)”;
- “Falta de conhecimento específico do proprietário/representante do produto (Product Owner)”;
- “Sobrecarga de trabalho do proprietário/representante do produto (Product Owner)”.

Os três primeiros tiveram o grau 4 com mais votos e o último teve o grau 3 com mais votos, conforme podem ser vistos em seguida nos Gráficos 7, 8, 9 e 10 que já foram mencionados anteriormente. Isso resulta em uma concordância com os autores da revisão sistemática, mostrando que o Product Owner é uma fonte potencial de desafios de Engenharia de Requisitos.

(b) Dependência do proprietário/representante do produto (Product Owner) como ponto único para coletar os requisitos.

31 respostas

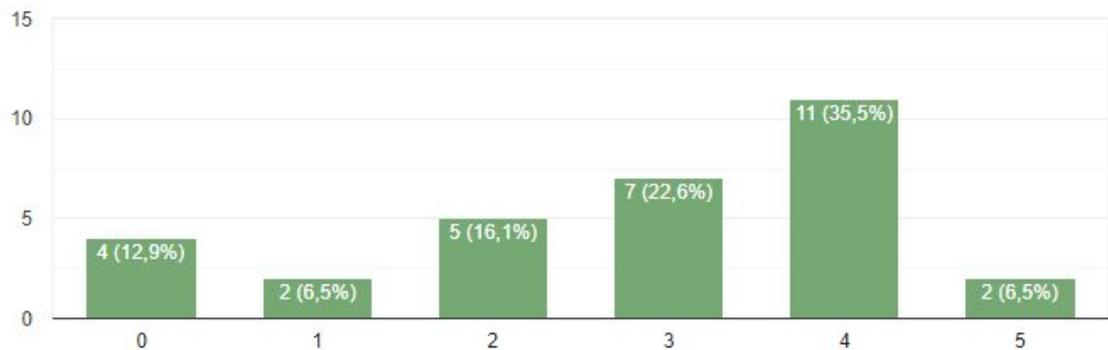


Gráfico 7: Percentual dos entrevistados conforme o desafio específico 2  
(Fonte: O Autor)

(d) Disponibilidade insuficiente do proprietário/representante do produto (Product Owner).

31 respostas

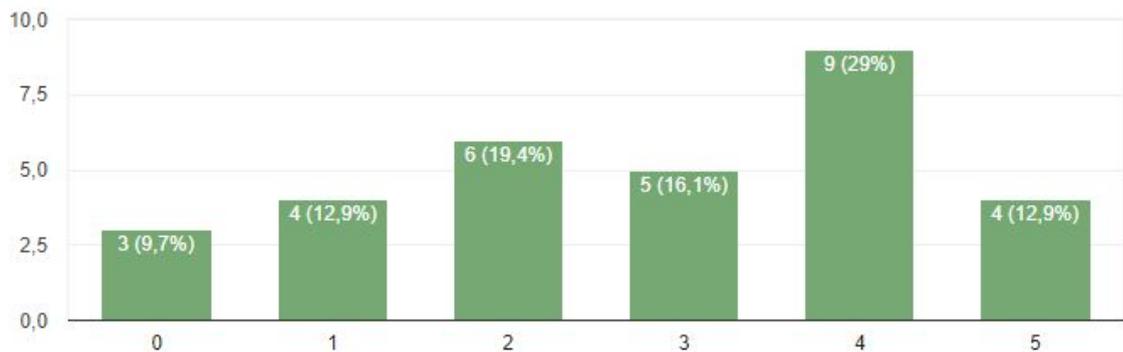


Gráfico 8: Percentual dos entrevistados conforme o desafio específico 4  
(Fonte: O Autor)

(g) Falta de conhecimento específico do proprietário/representante do produto (Product Owner).

31 respostas

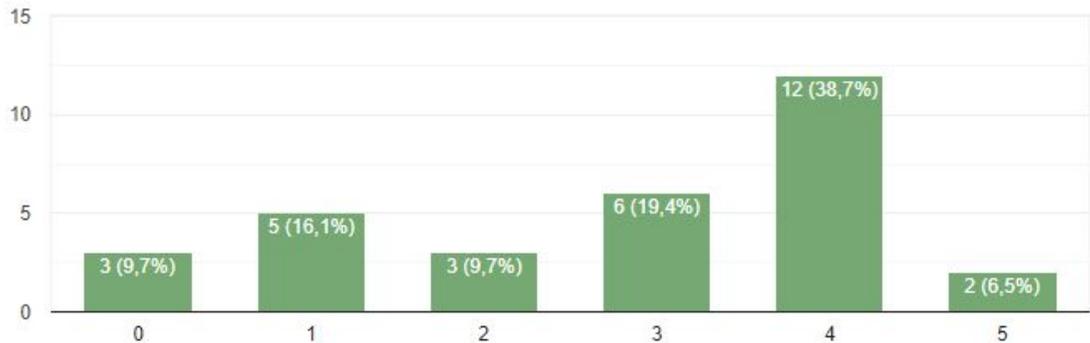


Gráfico 9: Percentual dos entrevistados conforme o desafio específico 7  
(Fonte: O Autor)

(i) Sobrecarga de trabalho do proprietário/representante do produto (Product Owner).

31 respostas

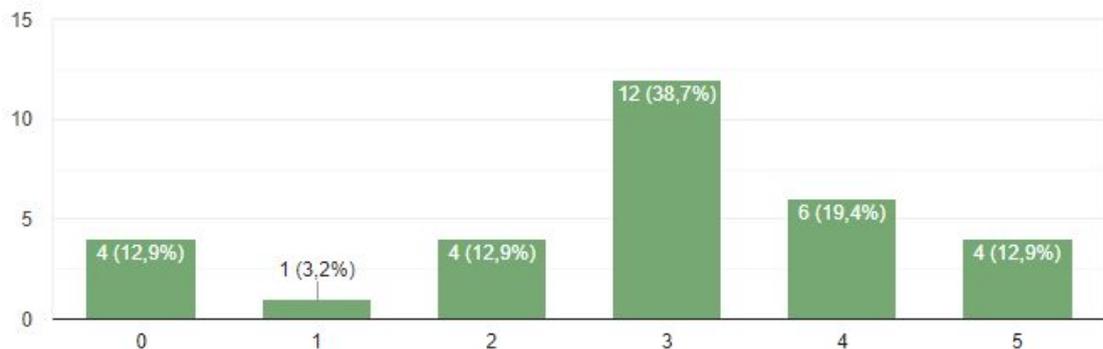


Gráfico 10: Percentual dos entrevistados conforme o desafio específico 9  
(Fonte: O Autor)

Referente a RQ3, que questionava sobre quais as soluções existentes para lidar com Requisitos de Qualidade negligenciados em Engenharia de Requisitos ágil em geral (não apenas em ALSA), conforme a literatura RE, não foi abordado nesta pesquisa pelo motivo de que a maioria das soluções propostas são criações de novos artefatos. Também foi questionado que essas novas soluções não foram consideradas como propostas ágeis, de acordo com o grau de agilidade que foi avaliado com base na presença de cinco elementos-chave: flexibilidade, velocidade, simplicidade, aprendizagem e capacidade de resposta.

Vale ressaltar que os autores do artigo analisado mencionaram o seguinte questionamento: "Se uma solução para cada tipo de QRs é proposta e adotada, acabamos com um grande número de abordagens. Ampliação ágil com ferramentas, tipos e/ou práticas necessárias para integrar cada tipo de QRs vai certamente resultar em uma abordagem pesada que não atende aos princípios ágeis definidos no manifesto ágil".

#### 4.3 Considerações finais

Nesta seção, pôde-se ter um conhecimento mais amplo sobre como está o panorama da relação dos requisitos de qualidade com o desenvolvimento de software ágil, juntamente com as técnicas utilizadas no processo de levantamento de requisitos no Brasil. Foi possível constatar as semelhanças e as divergências entre os resultados da pesquisa feita com uma RSL (ALSAQAF; DANEVA; WIERINGA, 2017) e o Survey realizado neste trabalho. Para que desta forma os leitores possam ter a oportunidade de saber a opinião dos entrevistados e a realidade da relação do uso de requisitos de qualidade em projetos ágeis no Brasil, com isso os profissionais terão o conhecimento dos aspectos positivos e negativos para que possam adaptar da melhor forma na sua aplicação prática.

### 5. Conclusão

A aplicação das metodologias ágeis no desenvolvimento de software tem avançado a cada ano para que a indústria de software tenha a capacidade de acompanhar a rápida procura do mercado. Foi constatado, na presente pesquisa, a grande adoção, por parte dos profissionais, das técnicas de engenharia de requisitos juntamente com os métodos ágeis. No entanto, a relação com os requisitos de qualidade ainda tem muito a melhorar visto que, de acordo com os resultados do survey, foi possível verificar quais os desafios específicos enfrentados por práticas ágeis em relação aos Requisitos de Qualidade mais vivenciados pelos entrevistados.

Também foi possível constatar as práticas de Engenharia de Requisitos mais vivenciadas pelos entrevistados em projetos de desenvolvimento ágil de software para lidar com os Requisitos de Qualidade.

#### 5.1 Contribuições

A presente pesquisa proporcionou um panorama geral sobre o desenvolvimento de software ágil em consonância com as técnicas de engenharia de requisitos no contexto brasileiro juntamente com o uso dos requisitos de qualidade nesse contexto. Desta forma, pôde-se obter uma visão geral das técnicas mais utilizadas para as fases de elicitação de requisitos. Também foi possível adquirir um conhecimento sobre a vivência nos desafios específicos enfrentados por práticas ágeis em relação aos Requisitos de Qualidade.

Além disso, esta pesquisa permitiu obter dados sobre as práticas de Engenharia de Requisitos para lidar com Requisitos de Qualidade mais vivenciados em projetos de

desenvolvimento ágil de software. Também gerou conhecimento sobre o grau de formação e nível de experiência que os profissionais que utilizam tais práticas possuem.

Esta pesquisa visou ainda realizar um cruzamento de informações provenientes do Survey aplicado e da revisão sistemática da literatura realizada por outros autores, para aprofundar os conhecimentos sobre este assunto e apontar o estado da prática e os desafios enfrentados no mercado de TI brasileiro. Isto possibilita que a academia possa visualizar as lacunas presentes no mercado e venha a intervir, com os seus estudos, na realidade relatada.

## 5.2 Limitações e trabalhos futuros

O fator tempo pode ser considerado um dos fatores limitadores dessa pesquisa, pois o questionário aplicado teve um curto período para obter respostas, que ocorreu entre os dias 07 e 25 de novembro de 2018. Outro fator de limitação é a falta de disponibilidade das pessoas para responder o questionário de pesquisa, pois mesmo em um curto período de tempo, foram realizadas diversas ações de compartilhamento do questionário que serviu como produtor de insumos para esta investigação. De fato, o questionário foi distribuído em listas com centenas de pessoas da área de Engenharia de Software, incluindo contatos com empresas de desenvolvimento de software. O presente estudo ainda enfrentou o desafio de extrair as opiniões dos profissionais de diversas partes do país. No entanto, lamentavelmente, não foram obtidas respostas provenientes das regiões Centro-Oeste e Norte.

Ressalta-se ainda que os dados obtidos foram relevantes e ainda poderiam ser utilizados de diversas formas, como para saber quais técnicas são utilizadas no levantamento de requisitos por profissionais de um determinado grau de formação ou tempo de experiência, qual o perfil de profissionais que mais vivenciaram as práticas mais votadas de Engenharia de Requisitos para lidar com Requisitos de Qualidade, entre outros. Contudo, essa análise foge ao escopo e cronograma inicialmente traçado para este trabalho. Os dados qualitativos extraídos do questionário, em alguns casos, foram bastante gerais ou resumidos, impossibilitando assim obter alguma conclusão sobre o assunto.

Como sugestão para futuras pesquisas, pode-se ainda realizar estudos que estejam focados em apenas um desafio específico enfrentado por práticas ágeis em relação aos Requisitos de Qualidade ou sobre os quatro desafios que estão relacionados ao papel do *Product Owner*, que também foram os mais evidenciados nesta pesquisa. Isto possibilitaria que fossem gerados materiais base para uma possível viabilização da criação de uma técnica ou método que pudesse minimizar os desafios evidenciados tanto na RSL como no Survey. Trabalhos futuros incluem desenvolver/evoluir técnicas de especificação de requisitos para lidar com requisitos de qualidade em projetos ágeis.

## Referências

ALVES. Sérgio de R., ALVES. André L, **ENGENHARIA DE REQUISITOS EM METODOLOGIAS ÁGEIS**. Universidade Católica de Goiás, Goiânia.

ARRUDA, Darlan; SOARES, Rafael; VIEIRA, Dércio. **Engenharia de Requisitos: Um Survey realizado no Porto Digital, Recife/Brasil**. 2013. 14 f. Artigo - Curso de Pós Graduação em Engenharia da Computação, Universidade de Pernambuco, Recife, 2013.

BECK, Kent et al. **Manifesto para o desenvolvimento ágil de software**. Disponível em: <<http://agilemanifesto.org>>. Acesso em: 04 dez. 2018.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. 176 p.

GROFFE, Renato Jose. **Desenvolvimento ágil com Scrum: uma visão geral**. Disponível em: <<https://www.devmedia.com.br/desenvolvimento-agil-com-scrum-uma-visao-geral/26343>>. Acesso em: 03 dez. 2018.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS – IEEE. **IEEE Guide for Software Requirements Specifications**. New York: IEEE, 1984.

Mairiza, D., Zowghi, D. and Nurmuliani, N. 2010. **An Investigation into the Notion of Non-Functional Requirements**. In Proceedings of the 2010 ACM Symposium on Applied Computing, Vancouver, Canada pp. 311-317

MEDEIROS, Juliana Dantas Ribeiro Viana de. **AN APPROACH TO SUPPORT THE REQUIREMENTS SPECIFICATION IN AGILE SOFTWARE DEVELOPMENT**. 2017. 180 f. Tese (Doutorado) - Curso de Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2017.

**METODOLOGIA Ágil, Dinamismo no Desenvolvimento de Software**. [2016] Disponível em: <<http://metodologiaagil.com/>>. Acesso em: 15 set. 2018.

Opservices, (2015) **Quais os benefícios das metodologias ágeis para a gestão de TI?** Disponível em: <<https://www.opservices.com.br/beneficios-das-metodologias-ageis>> Acessado em 16 de set de 2018.

PRODANOV. Cleber Cristiano, (2013) **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico**. ISBN 978-85-7717-158-3, Editora: Feevale, Novo Hamburgo.

SOMMERVILLE, Ian. **Software engineering**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

TOTVS, Equipe. **Metodologia ágil: entenda como funciona do conceito à prática**. Disponível em: <<https://www.totvs.com/blog/metodologia-agil/>>. Acesso em: 04 dez. 2018.

VERSIONONE, **10th Annual State of Agile Survey**. Disponível em: <<https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf/>>. Acesso em: 15 set. 2018.

Wagner, S., et al. (2018) Status Quo in Requirements Engineering: A Theory and a Global Family of Surveys. arXiv preprint arXiv:1805.07951.

W. Alsaqaf, M. Daneva, and R. Wieringa, “**Quality requirements in large scale distributed agile projects – a systematic literature review**,” in Requirements Engineering: Foundation for Software Quality, P. Grünbacher and A. Perini, Eds. Cham: Springer International Publishing, 2017, pp. 219–234.

## **Apêndice I - Survey realizado**

### **Questionário para estudo exploratório sobre os Requisitos de Qualidade em Projetos Ágeis na indústria brasileira de desenvolvimento de software**

#### **Termo de Consentimento Livre e Esclarecido (TCLE)**

A sua participação ocorrerá por meio de questionário, baseado em questões elaboradas pelo pesquisador Nikollas Filgueiras da Silva. Não haverá nenhuma outra forma de envolvimento ou de comprometimento neste estudo. É assegurado que seu nome não aparecerá, sendo mantido o mais rigoroso sigilo através da omissão total de quaisquer informações que permitam sua identificação.

Sua participação é voluntária e, a qualquer momento, você poderá pedir a retirada de seus dados desta pesquisa, bastando, para isso, entrar em contato com o pesquisador responsável pelo projeto através do seguinte e-mail: [nfs2@cin.ufpe.br](mailto:nfs2@cin.ufpe.br).

O resultado da pesquisa será divulgado na célula de pesquisa DARE (Dating with Advanced Requirements Engineering) do grupo ASSERTLAB da Universidade Federal de Pernambuco, além de ser utilizado como insumo para o Trabalho de Conclusão de Curso do pesquisador mencionado acima, podendo também ser apresentado em encontros ou em revistas científicas. Entretanto, este estudo mostrará apenas os resultados obtidos como um todo, sem revelar seu nome, instituição a qual pertence ou qualquer informação que esteja relacionada com a sua privacidade.

#### **Sobre o Termo de Consentimento Livre e Esclarecido (TCLE)**

( ) Fui informado (a) dos objetivos do Estudo exploratório sobre os Requisitos de Qualidade em Projetos Ágeis na indústria brasileira de desenvolvimento de software, de maneira clara e detalhada, e esclareci minhas dúvidas. Concordo que os materiais e as informações obtidas relacionadas à minha pessoa poderão ser utilizados em atividades de natureza acadêmico-científica, desde que assegurada a preservação de minha identidade. Sei que a qualquer momento poderei solicitar novas informações e modificar minha decisão de participar, se assim o desejar, de modo que declaro que concordo em participar desse estudo.

**1. Qual o seu nome?**

**2. Qual o estado (UF) em que você trabalha atualmente?**

**3. Qual a sua formação acadêmica?**

( ) Curso Técnico

( ) Superior Incompleto

( ) Superior Completo

( ) Mestrado

( ) Doutorado

( ) Outro: \_\_\_\_\_

**4. Você tem quantos anos de experiência profissional?**

( ) 0 a 3 anos

( ) 3 a 5 anos

( ) 5 a 10 anos

( ) 10 a 15 anos

( ) Mais de 15 anos

**5. Em qual cargo você trabalha atualmente?**

( ) Desenvolvedor

( ) Tester/QA

( ) Project Manager

( ) Product Owner

( ) Scrum Master

( ) DevOps

( ) Designer/UX

( ) Analista de Negócios

( ) Analista de Requisitos

( ) Outro: \_\_\_\_\_

**6. Qual o tamanho da empresa em que você está trabalhando?**

- Micro (até 9 empregados)
- Pequena (de 10 a 49 empregados)
- Média (50 a 99 empregados)
- Grande (mais de 100 empregados)

**7. Você possui quantos anos de experiência com metodologias ágeis?**

- Não tenho experiência
- 1 a 3 anos
- 3 a 5 anos
- 5 a 10 anos
- 10 a 15 anos
- Mais de 15 anos

**8. Com que frequência você utiliza métodos ágeis no desenvolvimento de software?**

- Não utilizo metodologia ágil
- 1 projeto ao ano
- 2 a 5 projetos ao ano
- 6 ou mais projetos ao ano

**9. No processo de levantamento de requisitos, utilizando metodologias ágeis, quais das técnicas citadas abaixo você já utilizou:**

- Entrevista
- Brainstorm
- Questionário
- Grupo Focal
- JAD (Joint Application Design)
- Trawling
- Workshop
- Outro: \_\_\_\_\_

**10. De acordo com os métodos que você assinalou acima, cite os pontos positivos da utilização de cada um deles.**

---

**11. De acordo com os métodos que você assinalou acima, cite os pontos negativos da utilização de cada um deles.**

---

**Sobre os requisitos de qualidade em projetos de desenvolvimento ágil de software**

**Requisitos de Qualidade são os requisitos que descrevem as qualidades do sistema (por exemplo, requisitos de desempenho, usabilidade, segurança, disponibilidade, entre outros).**

**12. Para as afirmações apresentadas nos itens abaixo de (a) até (l), pontue de 0 a 5, onde:**

- 0 significa que a afirmação nunca foi vivenciada por vocês;
- 1 representa que você discorda completamente da afirmação;
- 2 representa que você discorda parcialmente da afirmação;
- 3 representa que você não tem certeza se discorda ou concorda com a afirmação;
- 4 representa que você concorda parcialmente com a afirmação;
- 5 representa que você concorda completamente com a afirmação.

**(a) Métodos ágeis não fornecem uma técnica amplamente aceita para reunir os Requisitos de Qualidade.**

0 1 2 3 4 5  
( ) ( ) ( ) ( ) ( ) ( )

**(b) Dependência do proprietário/representante do produto (Product Owner) como ponto único para coletar os requisitos.**

0 1 2 3 4 5  
( ) ( ) ( ) ( ) ( ) ( )

**(c) Incapacidade de histórias de usuários para documentar os Requisitos de Qualidade.**

0 1 2 3 4 5  
( ) ( ) ( ) ( ) ( ) ( )

**(d) Disponibilidade insuficiente do proprietário/representante do produto (Product Owner).**

0 1 2 3 4 5  
( ) ( ) ( ) ( ) ( ) ( )

**(e) A validação dos Requisitos de Qualidade ocorre tarde demais no processo.**

0 1 2 3 4 5

**(f) Falta de mecanismos de rastreabilidade de requisitos.**

**0 1 2 3 4 5**

**(g) Falta de conhecimento específico do proprietário/representante do produto (Product Owner).**

**0 1 2 3 4 5**

**(h) Análise insuficiente dos requisitos.**

**0 1 2 3 4 5**

**(i) Sobrecarga de trabalho do proprietário/representante do produto (Product Owner).**

**0 1 2 3 4 5**

**(j) Ignorar os requisitos de arquitetura previsíveis.**

**0 1 2 3 4 5**

**(k) Incapacidade de histórias do usuário para documentar dependências de requisitos.**

**0 1 2 3 4 5**

**(l) Foco no fornecimento de funcionalidade ao custo da flexibilidade da arquitetura.**

**0 1 2 3 4 5**

**13. Para lidar com Requisitos de Qualidade, quais dessas práticas de Engenharia de Requisitos você já vivenciou em projetos de desenvolvimento ágil de software?**

- ( ) Comunicação face a face: Sessões de comunicação entre a equipe de desenvolvimento e o Dono/Representante do Produto (Product Owner) para identificar os requisitos.
- ( ) Os requisitos emergem durante o processo de desenvolvimento em vez de serem definidos antecipadamente.
- ( ) Priorização frequente de requisitos: Prioriza continuamente as histórias do usuário com base nas mudanças dos valores do negócio.
- ( ) Histórias de usuários: Descrição breve e abstrata dos requisitos usados para incentivar a comunicação face a face.
- ( ) Preparação de produtos: Revisita continuamente o conjunto de histórias de usuários para remover histórias obsoletas, criar novas histórias e reavaliar sua prioridade relativa.
- ( ) História de entrega: História do usuário estendida com informações adicionais, tais como: especificação funcional, requisitos de qualidade, design de alto nível e cenários de teste.
- ( ) Wiki: Ferramenta on-line que pode ser acessada por todos os membros da equipe para comunicar as histórias do usuário.
- ( ) Prototipação evolutiva: Um protótipo que é construído iterativamente para ser finalmente usado na produção.
- ( ) Desenvolvimento orientado a testes (TDD): Os requisitos são primeiramente transformados em casos de teste e, em seguida, o código necessário é gravado para passar nos casos de teste.
- ( ) Integração contínua: todas as cópias dos produtos de todos os membros da equipe são mescladas várias vezes ao dia para validar o funcionamento correto do produto.
- ( ) Refatoração: pouco antes do final de cada iteração, a estrutura interna do produto é reestruturada para melhorar a qualidade interna.
- ( ) Programação em pares: dois desenvolvedores trabalham juntos. Uma pessoa é responsável pela programação e a outra é responsável por controlar e validar o código
- ( ) Outro: \_\_\_\_\_

**14. De acordo com as práticas assinaladas acima, cite os pontos positivos e negativos de cada técnica utilizada.**

---

**15. Com relação ao cliente, quais casos abaixo já provocaram algum problema em projetos de desenvolvimento ágil de software nos quais você participou:**

- ( ) Expectativas dos clientes não foram atendidas
- ( ) Interação inadequada entre cliente e equipe técnica
- ( ) Usuários não sabem o que eles querem
- ( ) Pouca disponibilidade do cliente
- ( ) Outro: \_\_\_\_\_

**16. Com relação ao processo de desenvolvimento ágil de software, quais casos abaixo implicaram em problemas em projetos nos quais você atuou:**

- Validação dos requisitos sem a perspectiva do cliente
- Requisitos não acordados com a equipe de desenvolvimento
- Ineficiência na análise e inspeção dos requisitos
- Pouca disponibilidade de ferramentas específicas para Engenharia de Requisitos Ágil
- Pouca reusabilidade de requisitos
- Overscoping
- Outro: \_\_\_\_\_

**17. Você tem alguma sugestão/comentário em relação ao questionário? Qual?**

---