



Universidade Federal de Pernambuco
Centro de Informática
Graduação em Ciência da Computação

**AGILE TESTING: UMA ANÁLISE DA PRÁTICA
NO PORTO DIGITAL DO RECIFE**

Trabalho de Graduação

JOÃO GUILHERME FARIAS DUDA

ORIENTADOR: ALEXANDRE VASCONCELOS (amlv@cin.ufpe.br)

COORIENTADOR: IVALDIR DE FARIAS JUNIOR (ihfj@cin.ufpe.br)

Recife, Junho de 2017

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO GUILHERME FARIAS DUDA

“AGILE TESTING: UMA ANÁLISE DA PRÁTICA NO
PORTO DIGITAL DO RECIFE”

TRABALHO DE GRADUAÇÃO APRESENTADA À
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO
DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE
PERNAMBUCO COMO REQUISITO PARCIAL PARA
OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA
COMPUTAÇÃO.

ORIENTADOR: ALEXANDRE VASCONCELOS
COORIENTADOR: IVALDIR DE FARIAS JUNIOR

RECIFE, JUNHO/2017

Agradecimentos

Primeiramente, dedico este trabalho e declaro minha gratidão a minha mãe, Rute Farias, e a minha irmã, Beatriz Farias. Até onde a memória alcança, elas sempre estavam presentes, em apoio, cooperação e entendimento.

Em segundo lugar, quero agradecer ao Centro de Informática. O fantástico corpo docente proporcionou a mim experiências e aprendizados fundamentais para meu crescimento profissional e pessoal. Em especial, agradeço aos meus orientadores, professor Alexandre Vasconcelos e Ivaldir Junior, pelo apoio, direcionamento, imensa compreensão e críticas durante o desenvolvimento desta pesquisa. Também agradeço meus colegas de curso, pelo companheirismo, ajuda e caronas, semestre a semestre.

Em igual tamanho, presto minha gratidão ao CESAR, onde as motivações para essa pesquisa nasceram e onde tive oportunidade de expandir minha visão profissional, pondo em prática minhas ideias e aprendendo com trabalho diário.

Resumo

As metodologias de desenvolvimento ágil de software vem se tornando o padrão na indústria, devido às suas capacidades intrínsecas de enfrentar dificuldades comuns no desenvolvimento comercial de software, como a constante mudança de requisitos, integração de profissionais com diferentes habilidades, limitação de orçamento e necessidade de rápido feedback sobre o que é desenvolvido.

Neste contexto, a mentalidade ágil trouxe aos processos de teste e validação de software mudanças fundamentais em relação às metodologias tradicionais. Em um projeto ágil, as atividades de teste ocorrem desde a concepção dos requisitos até a validação do produto com o usuário final e, devido a isso, são executadas por vários perfis de profissionais envolvidos em um projeto de software. Ademais, o ato de testar estende-se da simples crítica ao produto (encontrar defeitos, por exemplo) para ser uma forma de suportar o time no desenvolvimento (testes de regressão automáticos, por exemplo).

Este trabalho propõe entender o estado de testes em projetos ágeis no ambiente do Porto Digital do Recife, com o intuito de sugerir à indústria e à academia caminhos para melhorar a aplicação dessas metodologias. A pesquisa terá como questões principais:

- Quais práticas de testes ágeis são utilizadas?
- Quais dificuldades são encontradas para adoção da mentalidade ágil de teste de software?
- Quais benefícios são observados pela adoção da mentalidade ágil de teste de software?
- Qual o perfil das empresas que utilizam testes ágeis?

Para responder estas perguntas, primeiramente, será realizado um mapeamento sistemático da literatura relativa a testes ágeis no Brasil, com o intuito de entender quais são as técnicas utilizadas, e percepções sobre seus benefícios e desafios. Em seguida, baseado-se no resultado do mapeamento sistemático, será realizado uma pesquisa de campo do tipo *survey* com os profissionais de *software* do Porto Digital. Com isso, espera-se ter um retrato do estado da prática de testes em projetos ágeis neste pólo de tecnológico.

Palavras-Chave: teste de software, metodologias ágeis, mapeamento sistemático, pesquisa de campo, Porto Digital.

Abstract

Agile software development methodologies have become the standard in industry due to its intrinsic capabilities to face common difficulties in commercial software development such as the constant change of requirements, integration of professionals with different skills, budget constraints and the need for rapid feedback.

In this context, the agile mindset has brought to software testing and validation processes fundamental changes in relation to traditional methodologies. In an agile project, the test activities take place from the design of the requirements to the validation of the product with the end user and, because of this, are executed by professionals with different profiles and skillsets. In addition, the act of testing extends itself from simple criticism of the product (finding defects, for example) to be a way of supporting the team in development (automated regression tests, for instance).

This work proposes to understand the state of testing in agile projects in the Porto Digital of Recife, with the intent of suggesting industry and academia ways to improve the application of these methodologies. The research will have as main questions:

- Which agile testing practices are used?
- What difficulties are encountered in adopting agile software testing mentality?
- What benefits are observed by the adoption of agile software testing mentality?
- What is the profile of companies that use agile tests?

To answer these questions, a systematic mapping of the literature on agile testing in Brazil will be carried out, in order to understand the techniques in practice, and perceptions about its benefits and challenges. Then, based on the result of the systematic mapping, a survey will be executed with the software professionals who work on Porto Digital. With this, it's expected that to have a picture of the state of the practice of testing in agile projects in this technological hub.

Keywords: *software testing, agile methodologies, systematic mapping, field research, Porto Digital.*

Lista de Figuras

Figura 1 – Quadrantes de Testes Ágeis, Fonte: Crispin e Gregory [13].....	8
Figura 2 – Fluxo do Test-Driven Developmentt, Fonte: Gomes [53].....	10
Figura 3 – Processo de Busca de Trabalhos Primários.....	21
Figura 4 – Exemplo de fichamento de estudo primário.....	23
Figura 5 – Resultados da Busca na Literatura.....	25
Figura 6 – Distribuição de estudos primários por ano de publicação.....	25
Figura 7 – Distribuição de estudos primários por quadrante de testes ágeis relacionado.....	26
Figura 8 – Grafo de Colaboração por Estados.....	27
Figura 9 – Técnicas relativas ao Primeiro Quadrante de Testes Ágeis.....	29
Figura 10 – Técnicas relativas ao Segundo Quadrante de Testes Ágeis.....	30
Figura 11 – Técnicas relativas ao Terceiro Quadrante de Testes Ágeis.....	31
Figura 12 – Técnicas relativas ao Quarto Quadrante de Testes Ágeis.....	32
Figura 13 – Experiência com projetos de software.....	36
Figura 14 – Experiência em projetos de software que utilizam metodologias ágeis.....	37
Figura 15 – Distribuição de respondentes por papel em projetos ágeis.....	38
Figura 16 – Número de funcionários nas empresas.....	39
Figura 17 – Contexto de desenvolvimento da empresa.....	40
Figura 18 – Tamanho das equipes ágeis.....	41
Figura 19 – Área tecnológica dos projetos ágeis.....	42
Figura 20 – Modelo de maturidade de projetos ágeis.....	43
Figura 21 – Projetos com equipe de teste dedicadas.....	44
Figura 22 – Distribuição geográfica das equipes de teste.....	45
Figura 23 – Áreas do projeto que utilizaram metodologias ágeis.....	45
Figura 24 – Distribuição das técnicas relativas ao Primeiro Quadrante de Testes Ágeis.....	47
Figura 25 – Distribuição dos papéis relativos ao Primeiro Quadrante de Testes Ágeis.....	48
Figura 26 – Frequência de utilização das técnicas relativas ao Primeiro Quadrante de Testes Ágeis.....	49
Figura 27 – Etapa de projeto de uso das técnicas relativas ao Primeiro Quadrante de Testes Ágeis.....	49
Figura 28 – Distribuição das técnicas relativas ao Segundo Quadrante de Testes Ágeis.....	50
Figura 29 – Distribuição dos papéis relativos ao Segundo Quadrante de Testes Ágeis.....	51
Figura 30 – Frequência de utilização das técnicas relativas ao Segundo Quadrante de Tetes Ágeis.....	52
Figura 31 – Etapa de projeto de uso das técnicas relativas ao Segundo Quadrante de Testes Ágeis.....	52
Figura 32 – Distribuição das técnicas relativas ao Terceiro Quadrante de Testes Ágeis.....	53
Figura 33 – Distribuição dos papéis relativos ao Terceiro Quadrante de Testes Ágeis.....	54
Figura 34 – Frequência de utilização das técnicas relativas ao Terceiro Quadrante de Testes Ágeis.....	55
Figura 35 – Etapa de projeto de uso das técnicas relativas ao Terceiro Quadrante de Testes Ágeis.....	56
Figura 36 – Distribuição das técnicas relativas ao Quarto Quadrante de Testes Ágeis.....	57
Figura 37 – Distribuição dos papéis relativos ao Quarto Quadrante de Testes Ágeis.....	58
Figura 38 – Frequência de utilização das técnicas relativas ao Quarto Quadrante de Testes Ágeis.....	59

Figura 39 – Etapa de projeto de uso das técnicas relativas ao Quarto Quadrante de Testes Ágeis	60
Figura 40 – Qualidade dos produtos de projetos ágeis em comparação com não-ágeis.....	61
Figura 41 – Nível de engajamento das equipes de projetos ágeis em comparação com não-ágeis	62
Figura 42 – Desafios para implantação de testes ágeis.....	63

Lista de Tabelas

Tabela 1 – Critérios de Inclusão e Exclusão de Trabalhos Primários.....	22
Tabela 2 – Lista de estudos primários selecionados.....	77
Tabela 3 – Mapeamento das técnicas nos estudos primários selecionados.....	80

Lista de Siglas e Abreviações

ATDD	Acceptance Test Driven Development
BDD	Behavior-Driven Development
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
MSL	Mapeamento Sistemático da Literatura
MPS.BR	Melhoria de Processos do Software Brasileiro
MPT.BR	Melhoria do Processo de Teste Brasileiro
PUCRS	Pontifícia Universidade Católica do Rio Grande do Sul
PUG-PE	Python User Group - Pernambuco
QTA	Quadrantes de Testes Ágeis
RSL	Revisão Sistemática da Literatura
SBES	Simpósio Brasileiro de Engenharia de Software
SBQS	Simpósio Brasileiro de Qualidade de Software
SEKE	International Conference on Software Engineering and Knowledge Engineering
TDD	Test-Driven Development
TFA	Testes Funcionais Automáticos
UFAM	Universidade Federal do Amazonas
UFCE	Universidade Federal do Ceará
UFPE	Universidade Federal de Pernambuco
WAMPS	Workshop Anual do MPS
WASHES	Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software
WTDQS	Workshop de Teses e Dissertações em Qualidade de Software

Sumário

Resumo	iv
Abstract	vi
1 Introdução	1
1.1 Motivação.....	2
1.2 Objetivos.....	2
1.3 Metodologia.....	2
1.4 Estrutura do Trabalho de Graduação.....	3
2 Referencial Teórico	4
2.1 Introdução.....	4
2.2 Teste de Software.....	4
2.3 Movimento Ágil.....	5
2.3.1 Definição	5
2.3.2 História.....	6
2.3.3 Princípios e Valores.....	6
2.4 Testes Ágeis.....	7
2.4.1 Quadrantes de Suporte ao Time	8
2.4.2 Quadrantes de Crítica ao Produto.....	15
2.5 Considerações Finais.....	18
3 Mapeamento Sistemático da Aplicação de Testes Ágeis no Brasil	19
3.1 Introdução.....	19
3.2 Método de Pesquisa.....	19
3.2.1 Perguntas de Pesquisa	20
3.2.2 Estratégia de Busca.....	20
3.2.3 Critérios de Inclusão e Exclusão de Estudos Primários.....	22
3.2.4 Extração e Síntese de Dados	23
3.2.5 Visão Geral do Processo.....	23
3.3 Resultados	24
3.3.1 Resultados do Procedimento de Busca	24
3.3.2 Evidências da Aplicação de Testes Ágeis no Brasil.....	28
3.4 Considerações Finais.....	33
4 Testes Ágeis no Porto Digital do Recife	34
4.1 Introdução.....	34
4.2 Classificação da Pesquisa	34
4.3 O Questionário	35
4.4 Características da Amostra	35
4.4.1 Experiência em Projetos de Software	35
4.4.2 Papéis Desempenhados em Projetos de Software.....	37
4.4.3 Características das Empresas.....	38
4.4.4 Características dos Projetos e Equipes de Desenvolvimento	40
4.5 Práticas de Testes Ágeis.....	46

4.5.1	Práticas do Primeiro Quadrante de Testes Ágeis	46
4.5.2	Práticas do Segundo Quadrante de Testes Ágeis.....	50
4.5.3	Práticas do Terceiro Quadrante de Testes Ágeis.....	53
4.5.4	Práticas do Quarto Quadrante de Testes Ágeis	56
4.6	Percepções Sobre Testes Ágeis.....	60
4.7	Considerações Finais.....	63
5	Conclusão	65
5.1	Considerações Finais.....	65
5.2	Limitações.....	66
5.3	Trabalhos Futuros.....	67
	Referências	68
	Apêndice A - Script de seleção de estudos semi-automatizada	73
	Apêndice B – Lista de Estudos Primários	76
	Apêndice C – O Questionário	81

“I didn't fail the test, I just found 100 ways to do it wrong.”

Benjamin Franklin

1 Introdução

O objetivo da engenharia de *software* é desenvolver software de qualidade dentro de custos estabelecidos [1]. Para atingir tal objetivo, faz-se necessário a detecção de erros e falhas nas atividades de desenvolvimento. Testar é a atividade que busca detectar essas falhas de maneira sistemática. Dado o crescimento exponencial da dependência humana para com software e do alto risco associado ao comportamento inesperado e incorreto nos sistemas computacionais, é necessário dar destaque aos processos de teste na indústria.

Tradicionalmente, testar é a atividade de validação do software em relação a um conjunto completo e detalhado de requisitos, garantindo que o produto esperado tenha sido construído. Entretanto, projetos de software tem, por natureza, um forte fator de incerteza que impossibilita o planejamento prévio de todos os aspectos envolvidos nos processos de desenvolvimento. Essa característica é reconhecida em projetos que utilizam metodologias de desenvolvimento ágil. Em projetos ágeis, o teste é uma atividade que gera informações sobre o comportamento do produto, com o objetivo de conduzir os próximos passos a serem realizados.

Este novo paradigma exige dos profissionais envolvidos em teste de software novas habilidades e flexibilidade. Uma pesquisa promovida pela SauceLabs [2] com 732 profissionais mostrou que 89% deles adotam metodologias ágeis, mas apenas 32% se consideram totalmente imersos na mentalidade ágil.

1.1 Motivação

O trabalho de Melo e outros [3] estudou o desenvolvimento do movimento ágil no Brasil, identificando as principais técnicas implantadas, fatores de dificuldades e benefícios agregados pelo uso dessas metodologias. Entre as principais dificuldades encontradas, estavam as mudanças culturais organizacionais necessárias para adoção completa da mentalidade ágil. Segundo os autores, as empresas brasileiras de *software* têm seus processos ainda muito dirigidos a planos, o que dificulta a implantação de metodologias dirigidas a *feedbacks* rápidos e constante adaptação.

1.2 Objetivos

À luz da importância dos testes para projetos de software e a dificuldade dos profissionais de adaptar-se às metodologias ágeis, esse trabalho tem como objetivo entender o estado de aplicação das principais técnicas de testes ágeis no Porto Digital do Recife. Esta pesquisa visa responder as seguintes questões:

- Quais práticas de testes ágeis são utilizadas?
- Quais dificuldades são encontradas para adoção da mentalidade ágil de teste de software?
- Quais benefícios são observados pela adoção da mentalidade ágil de teste de software?
- Qual o perfil das empresas que utilizam testes ágeis?

1.3 Metodologia

Diante da importância de testes e da variedade de técnicas utilizadas em projetos ágeis, este trabalho visa entender a aplicação dessas técnicas no contexto do Porto Digital do Recife. Para atingir esse objetivo, a pesquisa será dividida em duas etapas:

- Um mapeamento sistemático da literatura, com o objetivo de levantar as principais práticas de teste de software em contextos ágeis no Brasil;

- Uma pesquisa de campo com profissionais que atuam no Porto Digital, visando obter a perspectiva dos profissionais de *software* do pólo sobre testes em projetos ágeis.

1.4 Estrutura do Trabalho de Graduação

Contando com o capítulo de introdução, este trabalho é organizado em cinco capítulos. São eles:

Capítulo 2 – Apresenta os principais conceitos ligados a teste de software em contextos ágeis, em especial, o modelo Quadrantes de Testes Ágeis.

Capítulo 3 – Apresenta o mapeamento sistemático da literatura relativa a testes ágeis no Brasil. Primeiramente, explana-se o processo de busca e seleção de trabalhos. Em seguida, sumariza-se os dados levantados no processo de mapeado sistemático.

Capítulo 4 – Apresenta os resultados da pesquisa de campo com profissionais de desenvolvimento de *software* do Porto Digital do Recife. Os dados foram analisados de forma a responder às perguntas de pesquisa propostas para o trabalho.

Capítulo 5 – Apresenta as considerações finais da pesquisa e recomendações de trabalhos futuros.

2 Referencial Teórico

2.1 Introdução

O presente capítulo tem como objetivo apresentar os conceitos básicos envolvidos no desenvolvimento desta pesquisa. Primeiramente, explana-se o conceito de teste e sua importância para os projetos de desenvolvimento de software. Em seguida, conceitua-se as principais características de metodologias ágeis de desenvolvimento de software. Encerrando, com o intuito de familiarizar o leitor com as principais técnicas de teste em projetos ágeis, é apresentado o modelo Quadrantes de Testes Ágeis.

2.2 Teste de Software

Como toda atividade realizada por seres humanos, desenvolvimento de software está sujeito a falhas. Metodologias para levantamento de requisitos e boas práticas de programação, como modularização, ajudam a produzir produtos mais robustos, porém, não eliminam a possibilidade de erros. Estima-se que o custo de reparo de uma falha de software que chega até o usuário final aumenta em até duas ordens de grandeza em relação à falhas precocemente corrigidas. Neste contexto, entre os processos de garantia de qualidade de software, estão os testes.

Testes formam o conjunto de atividades de desenvolvimento de software que buscam fornecer evidências da confiabilidade e qualidade do produto [4]. Testes tem como principais objetivos [5]:

- Verificar se o que foi implementado está de acordo com todos os requisitos do sistema;
- Reduzir os custos de manutenção corretiva e retrabalho;

- Garantir a satisfação do cliente com o produto desenvolvido;
- Produzir casos de teste que têm elevada probabilidade de revelar erros ainda não descobertos, minimizando tempo e esforço para tal;
- Indicar a qualidade e a confiabilidade do software com base na comparação do resultado dos testes com os resultados esperados;
- Verificar a correta integração entre os componentes de software.

Testes são partes essenciais não só dos processos de engenharia de software, mas também de negócio [6]. Testes geram informações que são usadas para tomada de decisão, ao fornecer uma visão realista da confiabilidade e qualidade do produto em questão. Como é impossível testar completamente qualquer software não-trivial [7], a decisão do tamanho de investimento nos processos de teste deve ser guiada pela criticidade e impacto do produto no mercado e sociedade. Em geral, um projeto de software destina 45% dos seus recursos para atividade de teste [8].

2.3 Movimento Ágil

2.3.1 Definição

Metodologias ágeis são abordagens para desenvolvimento de software que reconhecem que esse tipo de atividade é inerentemente empírica, em contraste com atividades consideradas definidas [9]. Atividades empíricas são caracterizadas pela dificuldade (ou impossibilidade) de predição de resultados dado um conjunto inicial de requisitos e contexto de execução [10]. Estes tipos de atividades exigem curtos ciclos de “inspeção-e-adaptação” e *feedback*, com o intuito de lidar com as mudanças imprevisíveis e conflitantes em demandas de usuários e *stakeholders* [11]. Dadas estas características obrigatórias para o sucesso de processos de desenvolvimento de software, metodologias ágeis baseiam-se na colaboração com o cliente, no desenvolvimento iterativo e incremental, aceitando mudanças como parte essencial do próprio processo [1].

2.3.2 História

As metodologias tradicionais de software colocam grande importância nas etapas de planejamento prévio, design e documentação. Essa abordagem é justificada para projetos de grande porte, com múltiplas equipes de desenvolvimento e em sistemas críticos; porém, quando aplicadas a projetos de menor porte e criticidade, estes processos baseados em planos tornam-se frustrantes [1]. Esta insatisfação levou um grupo de 17 especialistas a proporem um manifesto com o objetivo de “descobrir maneiras melhores de desenvolver software” [12].

2.3.3 Princípios e Valores

O Manifesto Ágil consiste de 4 valores e 12 princípios que compõem a essência de qualquer metodologia ágil.

- Valores

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

- Princípios

- Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado;
- Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento;
- Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente;
- Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo;
- Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto;
- Construa projetos em torno de indivíduos motivados;
- Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho;
- O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face;

- Software funcionando é a medida primária de progresso;
- Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
- Contínua atenção à excelência técnica e bom design aumenta a agilidade;
- Simplicidade – a arte de maximizar a quantidade de trabalho não realizado – é essencial;
- As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis;
- Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

2.4 Testes Ágeis

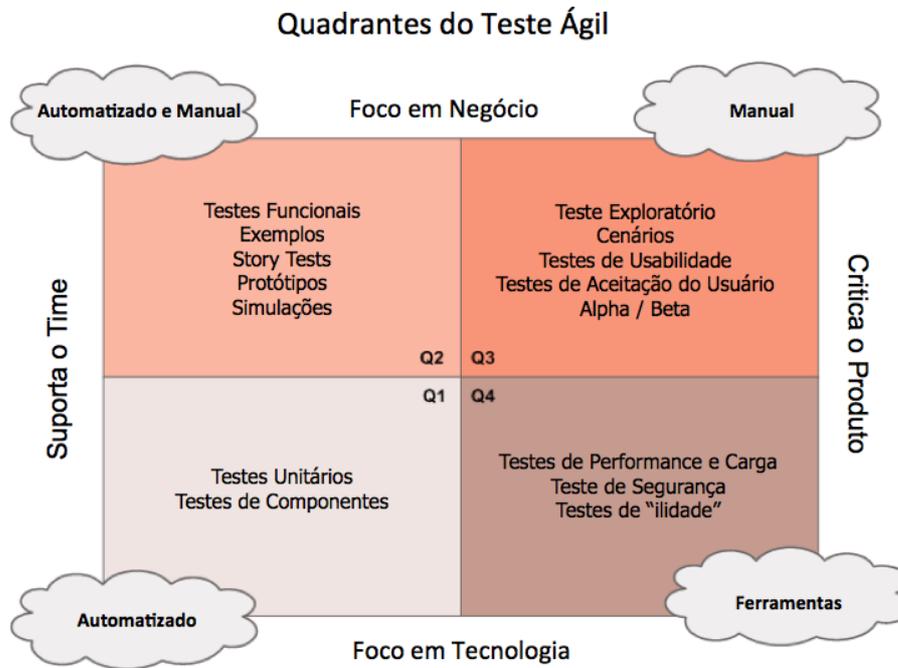
Assim como com os processos de levantamento de requisitos e desenvolvimento, o movimento ágil traz uma visão diferente de testes. Na luz de que é impossível planejar todo o projeto antecipadamente e devido à natureza altamente dinâmica e de iterações curtas de projetos ágeis, testes neste contexto tornam-se métodos para levantar informações sobre o comportamento do produto e direcionar a tomada de decisão do projeto. Appelo [52] afirma que projetos ágeis são dirigidos a testes, pois, em sua natureza, as etapas que um projeto passa dependem de resultados de ciclos de planejamento, implementação, validação e aprendizado.

Crispin e Gregory [13] propuseram um modelo de classificação de técnicas de teste em projetos ágeis denominado Quadrantes de Testes Ágeis (QTA). Este modelo divide testes sob duas perspectivas:

- Testes que buscam ajudar o time no desenvolvimento e testes que buscam criticar o produto desenvolvido;
- Testes que analisam o produto sob uma perspectiva do usuário, observando sua qualidade externa, e testes que analisam o produto sob uma perspectiva tecnológica, observando sua qualidade interna.

Combinando essas duas perspectivas, o modelo de QTA cria quatro categorias de testes, como mostrando na Figura 1.

Figura 1 - Quadrantes de Testes Ágeis



Fonte: Crispin e Gregory [13]

2.4.1 Quadrantes de suporte ao time

Os quadrantes Q1 e Q2 tem como objetivo suportar a equipe de desenvolvimento, fornecendo um melhor entendimento do produto em desenvolvimento. O quadrante Q1 engloba os testes que dirigem a construção da arquitetura do software, validando sua qualidade interna. Estes testes trazem confiança aos programadores que mudanças significativas no código, fato recorrente de projetos ágeis, podem ser realizadas sem regressões inesperadas em funcionalidade. O quadrante Q2 consiste de testes pertinentes à fase de levantamento de requisitos e a validação da qualidade externa do que está sendo desenvolvido. Neles, busca-se validar a compreensão das demandas dos usuários por

parte do time, certificando-se se o produto que pretende-se desenvolver resolve os problemas apresentados pelos clientes.

Técnicas do Quadrante 1

Testes Unitários

Testes unitários têm como objetivo validar o comportamento das menores partes testáveis de um programa, como um objeto ou um método [14]. Ao isolar partes do programa e validar seu comportamento, o programador atinge múltiplos objetivos:

- Provar que o comportamento daquela parte está correto;
- Prover um contrato de uso daquela parte do software, exemplificando o comportamento esperado sob diversos tipos de entradas;
- Realizar grandes mudanças em outras partes do código e validar se a unidade em questão teve seu comportamento alterado inesperadamente.

Testes unitários são os testes mais rápidos de serem executados e os mais precisos, por serem automatizados e focarem em pequenas partes da programa. Esses dois fatores são decisivos para detecção precoce de falhas, diminuindo, assim, os custos totais com correção de *bugs* do projeto [15].

Testes de Integração

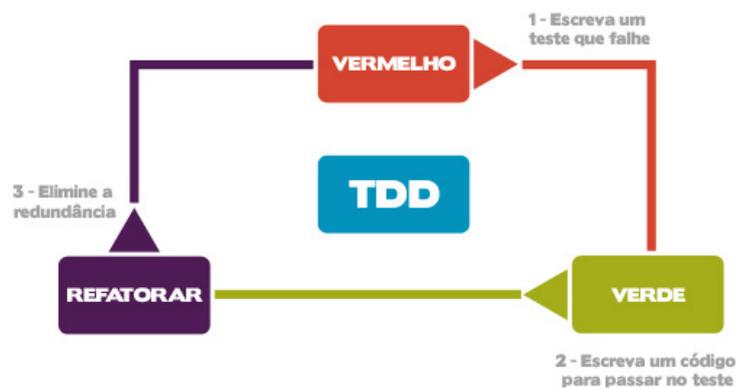
Testes de integração validam as interfaces de comunicação entre as partes do programa, como objetos, módulos e componentes [16]. Testes de integração têm como objetivo descobrir erros no uso dos contratos dos elementos validados isoladamente por testes unitários. Durante a integração de diversos componentes, surgem novas necessidades de uso e casos de uso para esses componentes. Assim, testes de integração

são parte essencial para solidificação do *design* de partes semanticamente completas de um sistema.

Test-Driven Development

Test-Driven Development (TDD) é uma metodologia de desenvolvimento que é baseada na repetição do fluxo diagramado pela Figura 2.

Figura 2: Fluxo do Test-Driven Development



Fonte: Gomes [53]

O fluxo de TDD inicia-se com a definição de um ou mais testes que descrevem uma funcionalidade. Em seguida, é desenvolvido o código mínimo para que estes testes passem. A terceira etapa é o refatoração do código desenvolvido, com a intenção de melhorar o *design* da arquitetura da aplicação.

Ao desenvolver pequenas partes do sistema de forma dirigida por testes, o desenvolvedor tem as seguintes vantagens [17]:

- Manter foco no desenvolvimento da funcionalidade em questão, evitando desvios de atenção para outras partes do sistema;
- Pensamento mais aprofundado no design da arquitetura do sistema, ao descrevê-la via testes como os componentes se comportam;
- Revisão constante dos requisitos, gerando assim discussões com os clientes sobre como o sistema deve se comportar.

Análise Estática de Código

Análise estática de código consiste na avaliação do código-fonte de um programa sem executá-lo. O principal objetivo da análise estática é extrair informações sobre a estrutura do código que podem indicar possíveis falhas e problemas de *design* arquitetural. Análise estática favorece melhorias de qualidade interna de software ao fornecer as métricas de qualidade de software [18]. Entre as possíveis métricas que podem ser extraídas estão:

- **Complexidade ciclomática:** Indicação quantitativa do número de caminhos linearmente independentes no código-fonte, geralmente medida a nível de método.
- **Acoplamento aferente:** Número de classes e interfaces de outros módulos que dependem do módulo em análise;
- **Acoplamento eferente:** Número de classes e interfaces de outros módulos dos quais o módulo em análise depende.

Análise estática fornece indicadores do impacto de reestruturações arquiteturais, permitindo aos desenvolvedores verificarem se os esforços de refatoração favoreceram a qualidade interna do produto.

Cobertura de testes

Cobertura de testes é um conjunto de métricas de software que avalia o grau pelo qual um programa é executado por uma suíte de testes. Geralmente expressa em porcentagem, alta cobertura de testes indica que um programa tem maiores porções de

seu código-fonte exercitados por testes. As métricas de cobertura diferenciam-se em relação ao objeto de contagem em questão. Entre as métricas associadas a cobertura de testes estão:

- Cobertura de funções;
- Cobertura de instruções;
- Cobertura de decisões;
- Cobertura de condições.

Programas com alta cobertura de testes tendem a ser menos sujeitos a *bugs* não detectados, tornando-os mais confiáveis [19]. Indústrias de alto risco, como aviação [20] e automóveis [21], utilizam cobertura de testes como métrica obrigatória de qualidade de seus produtos.

Programação em Pares

Programação em pares consiste na prática de desenvolvimento onde dois programadores trabalham colaborativamente no mesmo algoritmo, *design* ou tarefa, dividindo o mesmo ambiente de desenvolvimento [22]. O intuito dessa prática é a criação de *feedback* no intervalo de segundos, durante o desenvolvimento, sobre as decisões de implementação tomadas e qualidade do código. Ao trabalhar em pares, programadores discutem continuamente sobre seus modelos mentais do domínio e como eles são implementados em código. Ao ter que explicar para outra pessoa as abstrações enquanto elas são construídas, o programador reflete sobre a arquitetura do sistema e busca simplificá-la.

Cockburn e Williams [23] destacam como principais benefícios de programação em pares:

- Detecção precoce de erros;
- Melhoria de design e diminuição do tamanho de código-fonte;

- Maior velocidade para solução de problemas;
- Maior distribuição de conhecimento entre os membros do time;
- Maior engajamento dos membros das equipes.

Técnicas do Quadrante 2

Testes Funcionais

Teste funcional, também conhecidos como teste de caixa-preta, é um tipo de teste que verifica o comportamento do sistema contra um conjunto de documentos de *design* e requisitos. Diferentemente dos testes do quadrante 1, testes funcionais observam o sistema sob a perspectiva do usuário final, não focando métodos ou módulos, mas em pedaços de funcionalidade do sistema [24]. O objetivo de testes funcionais é validar que o produto implementa o entendimento das necessidades do usuário, determinando, assim, a qualidade externa do produto.

Boris define as seguintes etapas para execução de testes funcionais [25]:

- Identificação das funcionalidades que o programa deve executar;
- Criação dos dados de entrada, baseados nas especificações do sistema;
- Determinação das saídas esperadas, baseadas nas especificações do sistema;
- Execução dos casos de teste;
- Comparação dos resultados obtidos contra os esperados.

Devido ao fato que todos os casos de teste, dados de entrada e resultados esperados são desenvolvidos previamente à execução dos testes, testes funcionais geralmente são automatizados, permitindo mais velocidade e precisão na sua execução.

Prototipagem

Protótipo é uma versão inicial do sistema usado para demonstrar para usuários conceitos, experimentar opções de *design*, descobrir mais sobre o problema em questão e

suas possíveis soluções [1]. Em um projeto de software, os desenvolvedores podem usar protótipos para validar seu entendimento das necessidades dos usuários antes que a funcionalidade seja desenvolvida. Protótipos ajudam os usuários a:

- Criar novas ideias para requisitos;
- Observar erros e omissões nos requisitos propostos;
- Revisar a visão da funcionalidade dentro do contexto de interação com as outras funcionalidades do sistema.

Textos e diagramas não são meios expressivos o suficiente para simular interfaces gráficas [26]. Dado isso, um método comum de prototipagem de interfaces gráficas são os protótipos de papel. Esse método consiste em desenhos de mão livre que expressam os elementos fundamentais da interface do sistema. Com a ajuda de um membro da equipe de desenvolvimento, o usuário pode, então, simular o uso do software, encontrando problemas com o entendimento que ele e a equipe tinham sobre a funcionalidade. Devido ao baixo custo na produção e modificação de protótipos de papel, esse método torna-se especialmente efetivo para desenvolvimento rápido e iterativo de requisitos [27].

Inspeção de usabilidade

Inspeção de usabilidade é o nome de um conjunto de técnicas de análise de uma interface gráfica por parte de um avaliador. Essas técnicas têm como objetivo encontrar problemas de usabilidade nas fases iniciais de desenvolvimento, através da avaliação de protótipos ou especificações do sistema. Nielsen [28] lista os principais métodos de inspeção de usabilidade:

- **Avaliação heurística:** Consiste na avaliação dos elementos de uma interface gráfica contra um conjunto estabelecido de princípios de usabilidade (heurísticas);
- **Avaliação cognitiva:** Simula o processo de resolução de problemas do usuário em cada passo de uso do sistema, checando se os objetivos e memória do usuário simulado levam à próxima ação esperada no fluxo normal da funcionalidade avaliada.

- **Inspecção de consistência:** Durante inspeção de consistência, designers de outros projetos inspecionam uma interface com o objetivo de normalizar abstrações de conceitos entre vários sistemas.

2.4.2 Quadrantes de Crítica ao Produto

Os testes dos quadrantes Q3 e Q4 criticam o produto implementado, com o objetivo de gerar informações que guiem o progresso de desenvolvimento. O quadrante Q3 consiste de testes que investigam se o produto atende as necessidades do usuário, buscando encontrar falhas na etapa de levantamento de requisitos, como cenários omitidos e *designs* que não atendem as necessidades dos usuários. O quadrante Q4 analisa o comportamento do sistema em ambiente de produção. Testes de performance, carga e estresse validam se o sistema é capaz de sustentar-se em situações de uso extremo. Teste de segurança é uma etapa essencial para garantir que o produto e as informações que ele lida estão protegidos de ataques maliciosos.

Técnicas do Quadrante 3

Testes exploratórios

Teste exploratório é a atividade na qual um testador busca descobrir como um software funciona através de seu uso, analisando e questionando seu comportamento, com o objetivo de verificar sua qualidade e sugerir melhorias [24]. Testes exploratórios diferenciam-se de testes *ad-hoc* por seguirem um processo e terem objetivos bem definidos, apesar de também depender fortemente da criatividade e experiência dos testadores. Kaner e Bach [29] definem etapas comuns quando executam testes exploratórios:

1. Definição das áreas do sistema a serem avaliadas;
2. Análise de artefatos (especificações, protótipos, etc) relativo às áreas selecionadas;

3. Construção de um documento com o planejamento dos casos de testes a serem executados, especificando os recursos e tempo necessários os testes;
4. Preparação do ambiente de testes;
5. Execução dos casos de teste;
6. Anotação dos resultados e observações;
7. Atualização do documento de casos de testes baseada nas anotações, adicionando novos testes se necessário;
8. Se ainda houver validações a serem executadas, volta-se ao passo 4.

Testes de usabilidade

Testes de usabilidade são testes executados pelos usuários no produto desenvolvido. Através de observação da interação do usuários com o software, avaliadores coletam informações quantitativas e qualitativas sobre a facilidade de uso, quantidade de erros cometidos, motivos para confusão do uso. O objetivo principal dos testes de usabilidade é validar se o produto desenvolvido atende as necessidades dos usuários, de forma efetiva e eficiente. Assim como testes de inspeção, testes de usabilidade são essenciais por trazerem a visão do usuário final [30].

Myres [4] define um conjunto de considerações que devem ser respondidas por testes de usabilidade:

- A interface foi projetada de acordo com a inteligência, nível de educação e pressões contextuais dos usuários finais?
- As saídas do programa são significativas, não-abusivas e dentro da linguagem do usuário?
- O conjunto de interfaces apresenta consistência em seu *design*?
- O sistema apresenta os avisos necessários para garantir que o usuário só realizará operações críticas em pleno entendimento das consequências?
- O sistema apresenta *feedback* imediato para todas as operações?
- O sistema apresenta um número excessivo de opções, raramente usadas?

Testes de usabilidade também podem ser usados para avaliar um sistema em comparação com outros sistemas ou processos manuais [16], determinando se ele possui

vantagem competitiva. Por exemplo, perguntas como “O novo fluxo para compras no *ecommerce* diminui o tempo total que leva do momento que o usuário fecha seu carrinho para o momento que ele finaliza a compra?” podem determinar se o sistema pode trazer mais valor para o usuário final em relação a sua experiência atual.

Técnicas do Quadrante 4

Testes de Segurança

Testes de segurança formam um conjunto de atividades que buscam revelar falhas nos mecanismos de segurança de um software. Estes mecanismos garantem a integridade, confiabilidade e confidencialidade das informações e operações que permeiam um sistema computacional [31]. Dado isso, falhas nessas áreas diluem a qualidade geral de um software ao permitir que ataques maliciosos causem danos graves para os usuários.

Um processo de teste de segurança verifica a confiabilidade de um programa nos seguintes aspectos [13]:

- **Não-repudição:** Garantir que uma mensagem foi enviada e recebida pelos agentes que afirmam ter enviado e recebido a mensagem. Assim, garantindo que o remetente ou destinatário não poderão negar terem se envolvido na transação;
- **Autenticação:** Garantir a identidade de pessoas e artefatos;
- **Autorização:** Garantir que apenas usuários autorizados têm acesso a determinados serviços ou recursos;
- **Integridade:** Verificar que apenas pessoas e sistemas autorizados tem capacidade de alterar certas informações, com a intenção de garantir a corretude dos dados contidos no sistema.

Testes de Performance

Testes de performance buscam revelar o comportamento de um sistema em termos de responsividade e estabilidade sob diferentes situações de uso extremo. O principal

objetivo desses testes é fornecer informações sobre o sistema de modo a guiar a tomada de decisão dos *stakeholders*, pois gargalos e instabilidades podem ser críticos para os usuários finais [13].

De acordo com Molyneaux [32], testes de performance podem ser divididos em:

- **Teste de carga:** Analisa o comportamento do software sob diferentes níveis de carga, como acesso concorrentes ou alta demanda de banco de dados. Geralmente, busca-se investigar se há gargalos no tempo de resposta do sistema sob alta demanda;
- **Teste de estresse:** Avalia a capacidade do sistema de lidar com uma exigência crescente por recursos, determinando os limites superiores de uso que o sistema suporta;
- **Teste de resistência:** Investiga se o sistema tem a capacidade de suportar uso continuado dentro de uma carga esperada. Entre os problemas detectados por testes de resistência estão vazamentos de memória e degradação no uso otimizados dos recursos;
- **Teste de pico:** Avalia se o sistema é capaz de lidar com mudanças drásticas na carga de uso, observando por degradações de performance e tolerância a falhas.

2.5 Considerações Finais

Neste capítulo, apresentou-se a fundamentação teórica necessária para o entendimento deste trabalho. Primeiramente, foram conceituados Teste de Software e o Movimento Ágil. Em seguida, detalhou-se o modelo Quadrantes de Testes Ágeis, apresentando as principais técnicas de teste em projetos ágeis.

No capítulo seguinte, é apresentado um mapeamento sistemático da literatura relacionada a testes ágeis no Brasil.

3 Mapeamento Sistemático da Aplicação de Testes Ágeis no Brasil

3.1 Introdução

Após a introdução aos conceitos relacionados às técnicas teste de software em projetos que utilizam metodologias ágeis, é possível realizar um estudo sobre o estado da aplicação destas técnicas em um determinado contexto regional.

Este capítulo apresenta um mapeamento sistemático da literatura relacionadas às técnicas de teste ágeis no Brasil, buscando evidências de uso dessas técnicas na indústria brasileira. Esse tipo de estudo é particularmente interessante para este trabalho de graduação por prover em uma visão ampla de uma área de estudo [33]. Ademais, os resultados deste estudo, por serem baseados em experiências da indústria, podem ser comparados com a pesquisa de campo com profissionais do Porto Digital do Recife, destilada no próximo capítulo; provendo, assim, observações de avanços e lacunas da prática de testes em projetos ágeis nos dois ambientes.

3.2 Método de Pesquisa

Esta seção visa detalhar o processo de mapeamento da literária executado nesse trabalho de graduação. Seguindo as recomendações de Kitchenham e Charters [33], primeiramente foram definidas as perguntas de pesquisa que guiam as etapas de seleção, análise, extração e síntese de dados. A segunda etapa é a definição de um processo sistemático de aquisição de trabalhos primários candidatos. Em seguida, são definidos os

critérios para identificação de trabalhos primários que formarão o corpo de conhecimento que usaremos para responder as perguntas de pesquisa. A última etapa do processo de planejamento do mapeamento sistemático é a definição da estrutura de fichamento de dados e ferramentas de síntese destes.

3.2.1 Perguntas de Pesquisa

Este estudo tem como objetivo responder a seguinte Pergunta Primária de Pesquisa (PPP):

- **PPP:** Como testes ágeis são conduzidos no Brasil?

Com o objetivo de guiar o processo de mapeamento sistemático da literatura, as seguintes Perguntas de Pesquisa Secundárias (PPS):

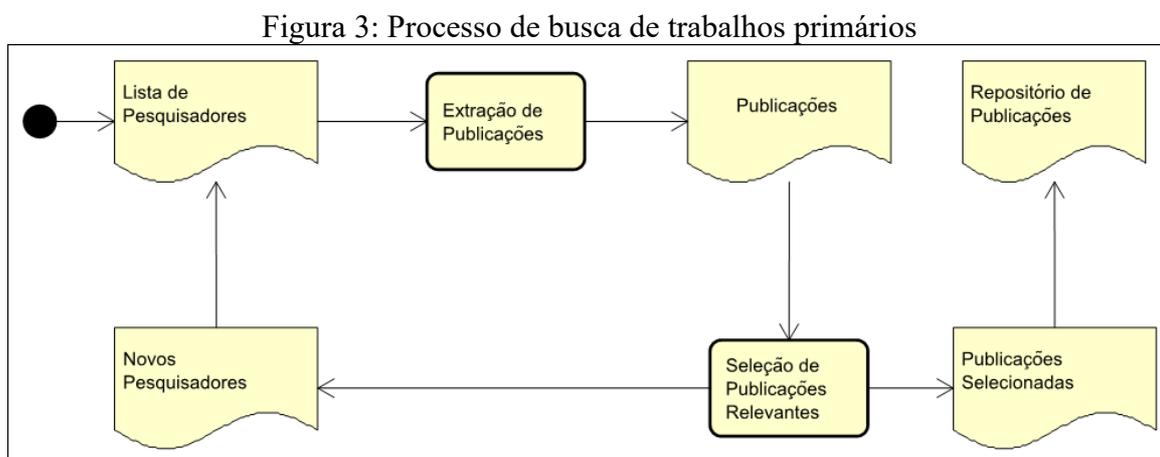
- **PPS1:** Quais técnicas relativas ao Primeiro Quadrante de Testes Ágeis (Q1) são usadas?
- **PPS2:** Quais técnicas relativas ao Segundo Quadrante de Testes Ágeis (Q2) são usadas?
- **PPS3:** Quais técnicas relativas ao Terceiro Quadrante de Testes Ágeis (Q3) são usadas?
- **PPS4:** Quais técnicas relativas ao Quarto Quadrante de Testes Ágeis (Q4) são usadas?

3.2.2 Estratégia de Busca

Seguindo as recomendações de Wohlin [34], foi realizada uma busca baseada no processo de *snowballing*. Entretanto, como este estudo é focado nas pesquisas realizadas no Brasil, buscou-se identificar pesquisadores atuantes no país na área de testes ágeis. A partir de uma lista de pesquisadores, foi conduzida uma busca de literatura no sistema de currículos Lattes¹ no dia 16 de Abril de 2017.

¹O "Currículo Lattes" é considerado um padrão brasileiro de informação sobre a produção científica e acadêmica de estudantes, professores, pesquisadores e profissionais envolvidos em ciência e tecnologia em geral. O banco de currículos pode ser acessado no site <http://lattes.cnpq.br/>

Para cada currículo analisado, foi extraída uma lista de trabalhos pertinentes a testes ágeis, divulgados entre os anos de 2012 e 2016, e um conjunto de autores que colaboraram nestes trabalhos. A nova lista de pesquisadores foi analisada pelo mesmo procedimento. O processo terminou quando nenhum novo pesquisador foi identificado. O processo é descrito pela Figura 3.



Fonte: Elaborado pelo autor

O processo da Figura 3 denominado “Extração de Publicações” foi realizado com o auxílio da ferramenta ScriptLattes [54]. Esta ferramenta agrupa as publicações de um conjunto de currículos da plataforma Lattes em um relatório bem-estruturado, possibilitando à análise de vários currículos em lote.

Utilizando este relatório como entrada, o processo da Figura 3 denominado “Seleção de Publicações Relevantes” foi realizado em 3 etapas. Primeiramente, o relatório do ScriptLattes foi analisado de forma semi-automatizada, utilizando um script (Apêndice A) em Python que pergunta ao usuário se o título do estudo é relevante ao mapeamento, salvando as informações do trabalho em um relatório caso a resposta seja

positiva. Caso o título do trabalho não for determinante para exclusão do processo de busca, ele é considerado para as etapas posteriores de análise. Os estudos selecionados servem de entrada para a segunda etapa do processo . Nela, o *abstract* de cada estudo é analisado de acordo com os critérios de inclusão e exclusão da Seção 3.2.3. A terceira etapa realiza uma análise do corpo completo dos estudos selecionados na segunda etapa, novamente aplicando os critérios de inclusão e exclusão da Seção 3.2.3.

3.2.3 Critérios de Inclusão e Exclusão de Estudos Primários

A Tabela 1 descreve os critérios de inclusão e exclusão de estudos primários. Para ser selecionado, um trabalho primário deve atender todos os critérios de inclusão (CI) e não pode satisfazer nenhum dos critérios de exclusão (CE).

Tabela 1: Critérios de Inclusão e Exclusão de Trabalhos Primários

Critérios de Inclusão	<ul style="list-style-type: none"> • Estudos que demonstrem a utilização de técnicas de testes ágeis • Estudos com validação na indústria • Estudos primários • Estudos revisados por pares • Estudos completos publicados em periódicos ou anais de congressos • Pesquisas qualitativas ou quantitativas
Critérios de Exclusão	<ul style="list-style-type: none"> • Estudos repetidos sem novas informações • Capítulos de livros, produção artística, patentes, resumos textos em jornais de notícias ou revistas • Estudos secundários ou terciários • Estudos limitados à aplicação educacional • Estudos não disponíveis na rede institucional da UFPE • Estudos não realizados no Brasil

Fonte: Elaborado pelo autor

3.2.4 Extração e Síntese de Dados

As informações de contexto, técnicas utilizadas e resultados foram extraídas em um formulário padrão. A Figura 4 mostra o fichamento de um estudo primário selecionado de acordo com os critérios da Seção 3.2.3.

Figura 4: Exemplo de fichamento de estudo primário

```
- titulo: "Processes versus people: How should agile software development maturity be defined?"
nome do revisor: "João Farias"
data: "23/04/2017"
inclusao:
  status: "Positivo"
  comentario: ""
dados:
  contexto:
    comentarios: ""
  avaliacao testes agels:
    beneficios: ""
    desafios: ""
    comentarios: ""
  modelos de maturidade:
    comentarios: "CMMI-DEV"
  quadrantes:
    Q1:
      tecnicas: "Pair Programming;Code Metrics;Static code analysis;Code review;Unit testing;Integration testing;TDD"
      ator: ""
    Q2:
      tecnicas: ""
      ator: ""
    Q3:
      tecnicas: ""
      ator: ""
    Q4:
      tecnicas: ""
      ator: ""
```

Fonte: Elaborado pelo autor

Os dados foram analisados utilizando scripts escritos em Python. Gráficos e tabelas foram criados utilizando a ferramenta LibreOffice.

3.2.5 Visão Geral do Processo

A pesquisa foi realizada em 3 etapas. A lista inicial de autores foi criada utilizando a lista de pesquisadores de metodologias ágeis no Brasil indicados por Goldman e Katayama [35], somando-se a um conjunto de autores que divulgaram trabalhos relevantes à pesquisa nas conferências SBES, SBQS, WAMPS, WASHES, WTDQS, entre os anos 2012 e 2016. A partir desta lista de autores, o processo de *snowballing* descrito na Seção 3.2.2 formou o banco de artigos inicial deste estudo. Em seguida, os

critérios de inclusão e exclusão descritos na Seção 3.2.3 foram aplicados nos títulos e resumos dos trabalhos; em caso de dúvida, os artigos foram considerados para análise mais detalhada nas etapas seguintes. Na terceira etapa, os artigos remanescentes foram analisados em seu completo, aplicando-se os mesmo critérios de inclusão e exclusão.

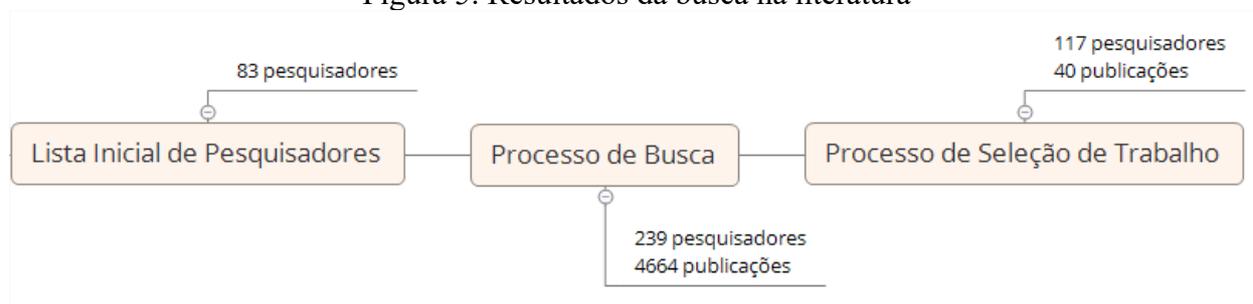
3.3 Resultados

Esta seção apresenta os resultados extraídos do processo de mapeamento sistemático da literatura explanado na Seção 3.2. A Seção 3.3.1 traz informações sobre o conjunto de trabalhos primários selecionados, como sua distribuição temporal de divulgação e relacionamento entre pesquisadores envolvidos. A Seção 3.3.2 responde às perguntas de pesquisa definidas na Seção 3.2.1.

3.3.1 Resultados do Procedimento de Busca

O processo de *snowballing* de autores resultou na análise de 239 pesquisadores, dos quais foram extraídos um total de 4664 estudos únicos. A análise de títulos e resumos reduziu o número de artigos com potencial de seleção para 223. Após análise de texto completo, o conjunto final de 40 estudos primários foi definido. A Figura 5 mostra os números de estudos resultantes de cada etapa. A lista de artigos selecionados está disponível no Apêndice A.

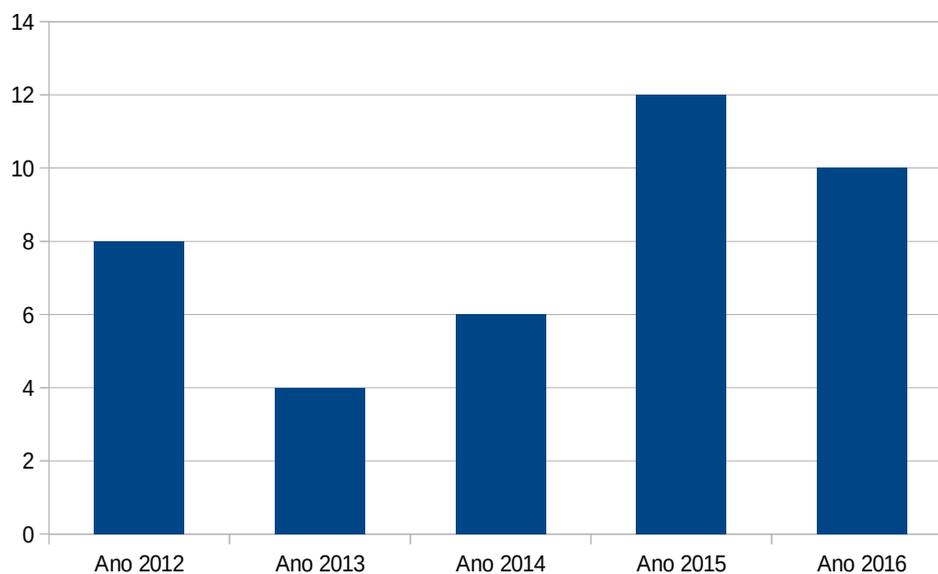
Figura 5: Resultados da busca na literatura



Fonte: Elaborado pelo autor

A Figura 6 mostra a distribuição dos trabalhos primários por ano de divulgação. Dos 40 estudos selecionados, 55% foram divulgados nos últimos 2 anos, mostrando o crescimento das experiências com técnicas de testes ágeis na comunidade acadêmica brasileira.

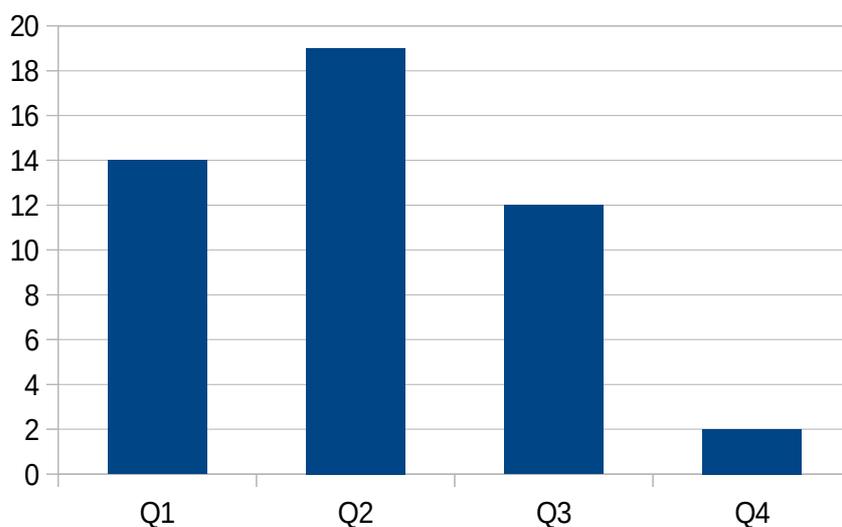
Figura 6: Distribuição de estudos primários por ano de publicação



Fonte: Elaborado pelo autor

Os meios de publicação mais utilizados desses estudos foram as conferências, simpósios e workshops, em especial, o Simpósio Brasileiro de Qualidade de Software (SBQS) e Conference on Software Engineering and Knowledge Engineering (SEKE). Os quadrantes de testes ágeis mais citados, como mostra a Figura 7, nos artigos foram o de suporte ao desenvolvimento (Q1 e Q2). Baseado nisso, podemos observar menos interesse no estudo de técnicas relacionadas a crítica do produto (Q3 e Q4). O Q3 é de importância para detecção de oportunidades de melhoria da experiência do usuário e validar se o produto atende às necessidades dos usuários. Enquanto o Q4 é decisivo para garantia da sustentabilidade do software em ambiente de produção, em especial em momentos de uso atipicamente intenso ou de ataques maliciosos.

Figura 7: Distribuição de estudos primário por quadrante de testes ágeis relacionado

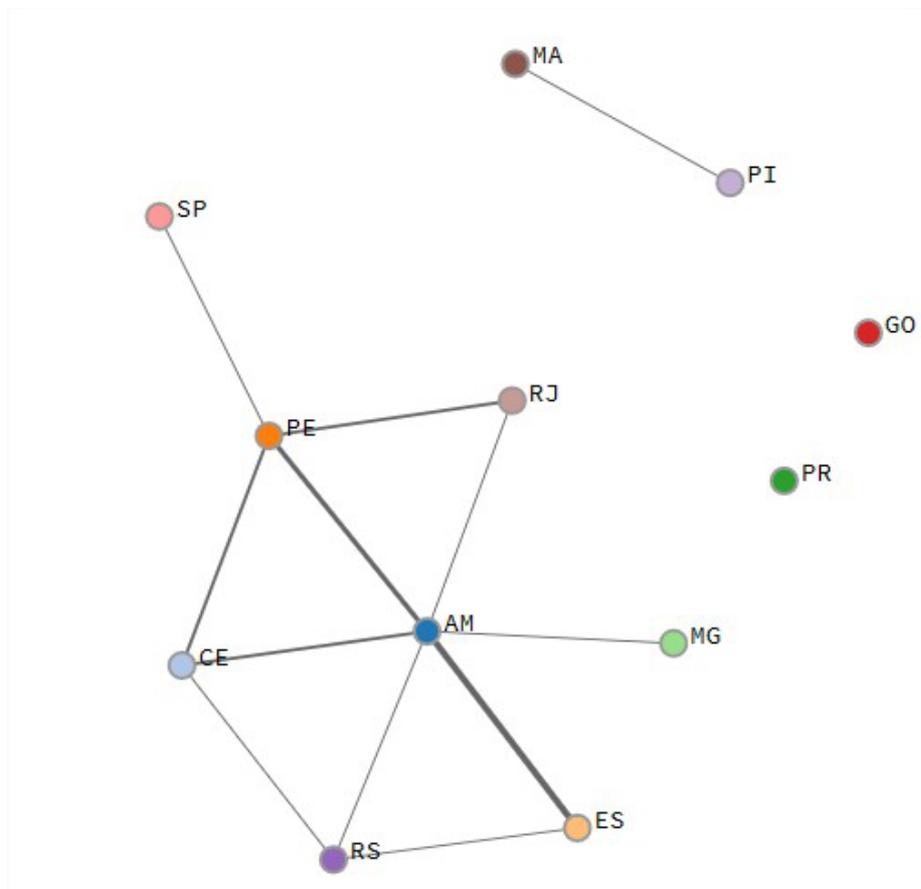


Fonte: Elaborado pelo autor

A maior parte das pesquisas selecionadas foram resultado de estudos na região Nordeste e Norte, em especial os estados de Pernambuco e Amazonas, seguidas da

regiões Sul e Sudeste. As universidades mais ativas foram a UFAM, UFPE, UFCE e PUCRS. Uma observação importante foi que apenas 22 pesquisadores (18% do total) são ligados a instituições não-acadêmicas, mostrando uma possível lacuna de incentivo a pesquisa e divulgação das experiências diretamente da indústria. A Figura 8 mostra a intensidade de colaboração entre pesquisadores de diferentes estados da federação.

Figura 8: Grafo de colaboração por estados



Fonte: Elaborado pelo autor

3.3.2 Evidências da Aplicação de Testes Ágeis no Brasil

Esta seção apresenta as evidências de uso de testes ágeis no Brasil. Para cada Pergunta de Pesquisa Secundária delineada na Seção 3.2.1, far-se-á uma sumarização das técnicas apresentadas nos trabalhos primários de forma qualitativa, dissertando sobre os resultados apresentados, e quantitativa, com a distribuição de frequência de suas citações.

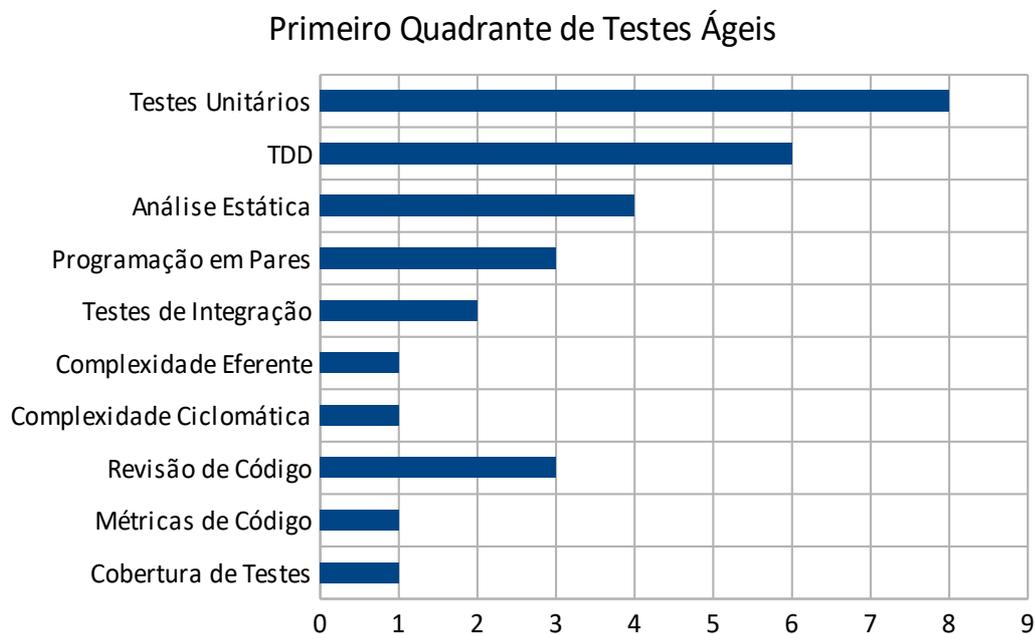
PPS1: Quais técnicas relativas ao Primeiro Quadrante de Testes Ágeis (Q1) são usadas?

Um total de 14 estudos utilizaram-se de técnicas de testes com objetivo de suportar a equipe desenvolvimento com foco em qualidade de código, como mostra a Figura 9. Estes estudos demonstraram o uso de 11 técnicas diferentes. A técnica mais utilizada foi o Teste Unitário, seguido de *Test-Driven Development* (TDD) e Análise Estática de Código.

O trabalho de Filho e outros [36] estudou o impacto da metodologia TDD em relação a severidade de erros detectados apenas após o processo de *building*. Nos projetos que se utilizaram de TDD, apenas 27% dos erros encontrados após o *building* foram considerados severos. Em projetos que não se utilizaram de TDD, 51% dos erros foram considerados severos.

Análise estática mostrou-se uma ferramenta para garantir a qualidade interna de software [37], permitindo aos times gerenciarem melhor seu débito técnico [38]. Esse controle da qualidade interna do software permite aos times realizarem mudanças sem causar danos inesperados ao sistema, adicionando constantemente valor ao produto [39].

Figura 9: Técnicas relativas ao Primeiro Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

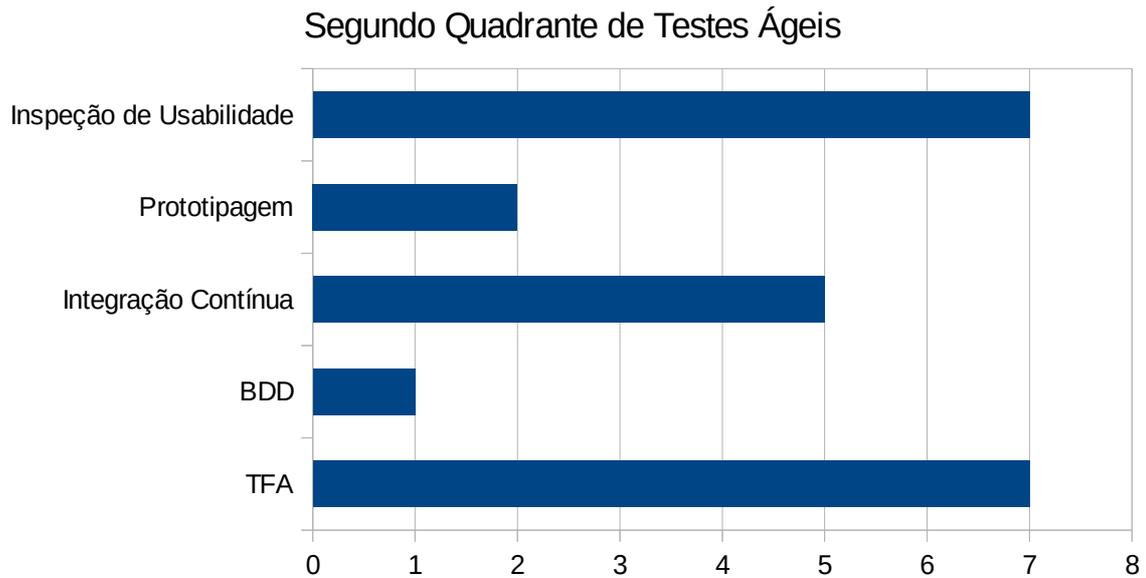
PPS2: Quais técnicas relativas ao Segundo Quadrante de Testes Ágeis (Q2) são usadas?

Um total de 19 estudos utilizaram-se de técnicas de testes com objetivo de suportar a equipe desenvolvimento com foco em negócio, como mostra a Figura 10. Estes estudos demonstraram o uso de 5 técnicas diferentes. As técnicas mais utilizadas foram os Testes Funcionais Automatizados e a Inspeção de Usabilidade.

Devido ao rápido *feedback* e capacidade de garantir a qualidade externa do software, testes funcionais automáticos (TFA) são fundamentais em projetos ágeis. De Castro e outros [40] desenvolveram uma abstração do *framework* Selenium para verificações em banco de dados, conseguindo diminuir o tempo de testes funcionais em

88%, quando comparado com testes manuais. TFA também demonstrou-se eficaz em contextos de aplicativos móveis [41] e softwares críticos [42].

Figura 10: Técnicas relativas ao Segundo Quadrante de Testes Ágeis



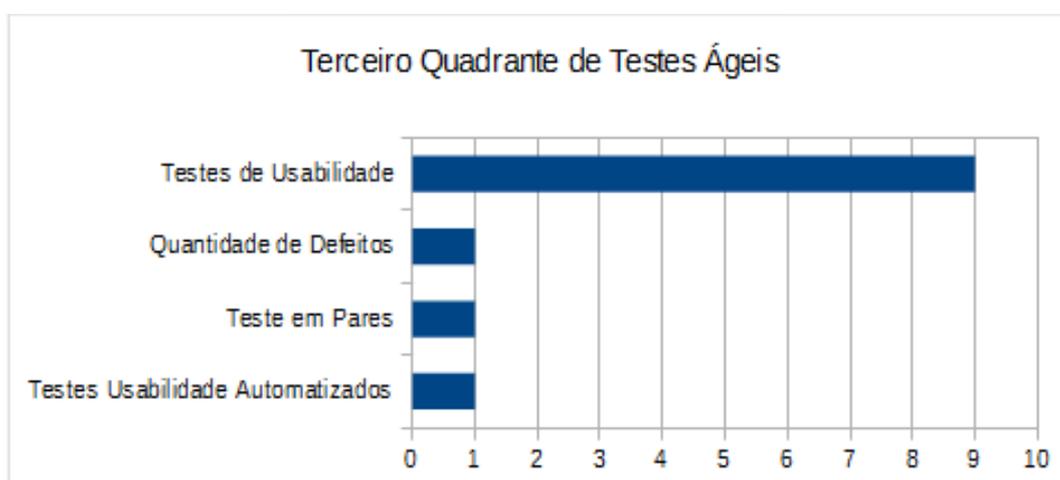
Fonte: Elaborado pelo autor

Os trabalhos De Vaz e outros [43] e Rivero e outros [44] definiram processos de inspeção de usabilidade, demonstrando como essa técnica é capaz de detectar falhas na fase de levantamento de requisitos. O uso de ferramentas que auxiliam o processo de inspeção mostrou-se eficaz, porém, também foram observadas diferenças significativas na quantidade e severidade dos defeitos quando comparados inspetores com diferentes níveis de experiência, reforçando a necessidade de valorização da habilidade humana no teste de software.

PPS3: Quais técnicas relativas ao Terceiro Quadrante de Testes Ágeis (Q3) são usadas?

Um total de 12 estudos utilizaram-se de técnicas de testes com objetivo de criticar o produto com foco em negócio, como mostra a Figura 11. Estes estudos demonstraram o uso de 4 técnicas diferentes. A técnica mais utilizada foi o Teste de Usabilidade.

Figura 11: Técnicas relativas ao Terceiro Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

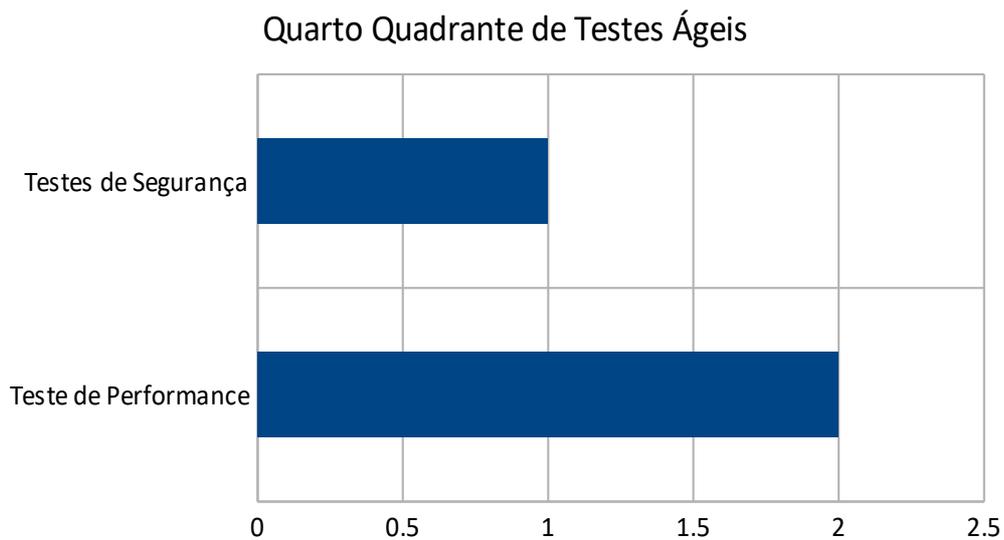
Testes de usabilidade tem como objetivo principal verificar se o produto desenvolvido é capaz de ser usado de forma eficaz pelo usuário final. O testador pode se basear em uma lista de princípios de usabilidade [45], utilizar ferramentas para detecção automática de falhas [46][47] e colher feedback dos usuários em forma de survey. Falcao e Soares [48] utilizaram-se de testes de usabilidade para validar vantagem competitiva de um produto. Foi analisado se um dispositivo baseado em gestos podia trazer melhorias de

efetividade no trabalho de profissionais de *design*, em comparação a ferramentas tradicionais baseadas em teclado e mouse. Após levantar uma lista de problemas de usabilidade e dados de velocidade e eficácia de uso, os pesquisadores mostraram que o dispositivo não atenderia seus usuários de forma efetiva.

PPS4: Quais técnicas relativas ao Quarto Quadrante de Testes Ágeis (Q4) são usadas?

Um total de 3 estudos utilizaram-se de técnicas de testes com objetivo de criticar o produto com foco tecnológico, como mostra a Figura 12. Estes estudos demonstraram o uso das técnicas de Teste Performance e Teste de Segurança.

Figura 12: Técnicas relativas ao Quarto Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

Os testes do quadrante Q4 observam o produto sob condições de produção, onde picos de uso e ataques maliciosos podem causar graves danos organizacionais. Entretanto, estes testes, por serem puramente técnicos, são muitas vezes delegados para

fases finais dos projetos [13]. Lucena e Tizzei [38] destacam os problemas causados por esse tipo de decisão:

“When the software is finally deployed in the real customer environment one often finds integration and performance issues which could be prevented earlier. Late software deployments also limit useful feedback for the development team and delays customer return of the investment.”

3.4 Considerações Finais

Este capítulo apresentou os resultados de um mapeamento sistemático da literatura relacionada a testes ágeis no Brasil. Um total de 40 trabalhos de autores que atuam no Brasil foram classificados de acordo com o modelo Quadrantes de Testes Ágeis. As técnicas de teste, seus contextos de uso, os benefícios e desafios associados a sua aplicação foram sumarizados.

O próximo capítulo apresenta uma pesquisa de campo realizada com profissionais do Porto Digital do Recife, com o intuito de entender como são aplicadas as técnicas de testes ágeis no projetos desenvolvidos no pólo em questão.

4 Testes Ágeis no Porto Digital do Recife

4.1 Introdução

Este capítulo apresenta os resultados de uma pesquisa de campo com profissionais da indústria de software do Porto Digital do Recife. O objetivo da pesquisa foi entender como os processos de testes em projetos ágeis ocorrem. Para isso, buscou-se identificar as principais técnicas de teste, o contexto de seu uso, além de benefícios e desafios associados.

4.2 Classificação da Pesquisa

De acordo com Gil [49], pesquisas sociais podem ser divididas em três categorias, de acordo com seus objetivos: exploratórias, explicativas e descritivas. Pesquisas exploratórias buscam criar uma familiaridade do leitor com o tema em questão. Pesquisas explicativas identificam fatores relacionados ou determinantes para ocorrências de certos fenômenos em contextos específicos. Por último, pesquisas descritivas visam descrever as características de determinada população ou fenômeno. A presente pesquisa classifica-se como descritiva. O método de pesquisa descritiva adotado, denominado levantamento de campo ou *survey*, diferencia-se de outros tipo de pesquisa, principalmente do método censo, pelo fato de examinar apenas uma amostra da população [50].

Técnicas quantitativas e qualitativas de análise de dados foram empregadas. As abordagens quantitativas foram utilizadas nas questões objetivas, com respostas

determinadas previamente pelos pesquisadores. Técnicas qualitativas foram usadas nas questões subjetivas, com o objetivo de entender a percepção do uso das práticas de testes ágeis nas empresas.

4.3 O Questionário

O elemento central de um *survey* é o questionário, ferramenta pela qual são coletados os dados de uma amostra da população. Esta pesquisa focou-se nos profissionais de desenvolvimento de software do Porto Digital do Recife que participaram de projetos que utilizaram-se de metodologias ágeis. O questionário (Apêndice B) foi divulgado para contatos em empresas, mídias sociais e listas de e-mails de universidades e grupos de praticantes, como o Grupo de Usuários de Python de Pernambuco (PUG-PE).

O questionário consiste de 35 questões, dos tipos subjetivas, objetivas, obrigatórias e não obrigatórias. O *survey* foi aberto ao público no dia 5 de Junho de 2017 e aceitou respostas até o dia 16 de Junho de 2017. Foram coletadas um total de 36 respostas.

4.4 Características da Amostra

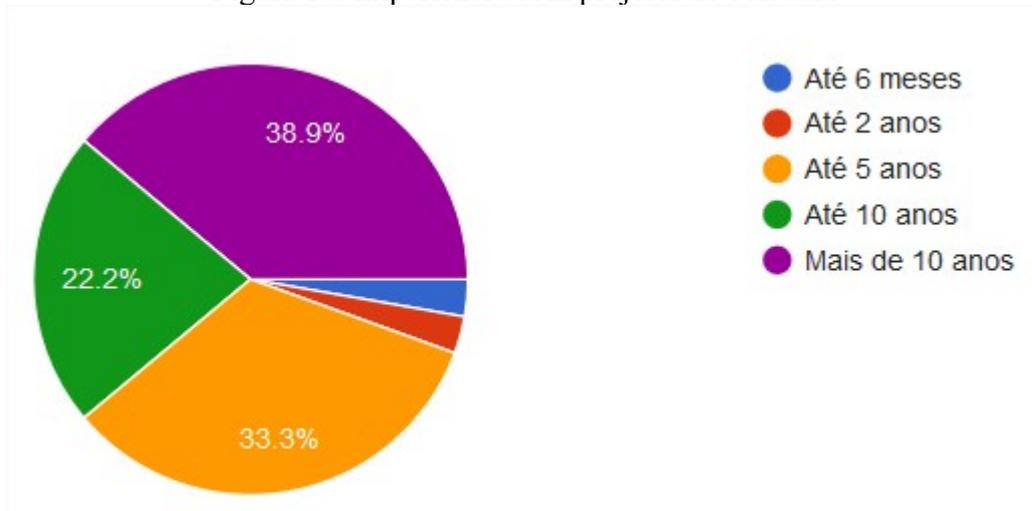
Após concordar com os termos de consentimento, o respondente é direcionado à segunda seção, denominada “Contexto de Projeto”, composta por 11 perguntas. O objetivo dessa seção do questionário é coletar informações sobre as características de projeto e organização em que o respondente aplicou testes ágeis.

4.4.1 Experiência em Projetos de Software

As duas primeiras perguntas da Seção 2 do questionário busca identificar o tempo de experiência do respondente em projetos de software em geral e em projetos ágeis.

A Figura 13 mostra a distribuição dos respondentes em relação ao tempo de experiência em projetos de software.

Figura 13: Experiência com projetos de software

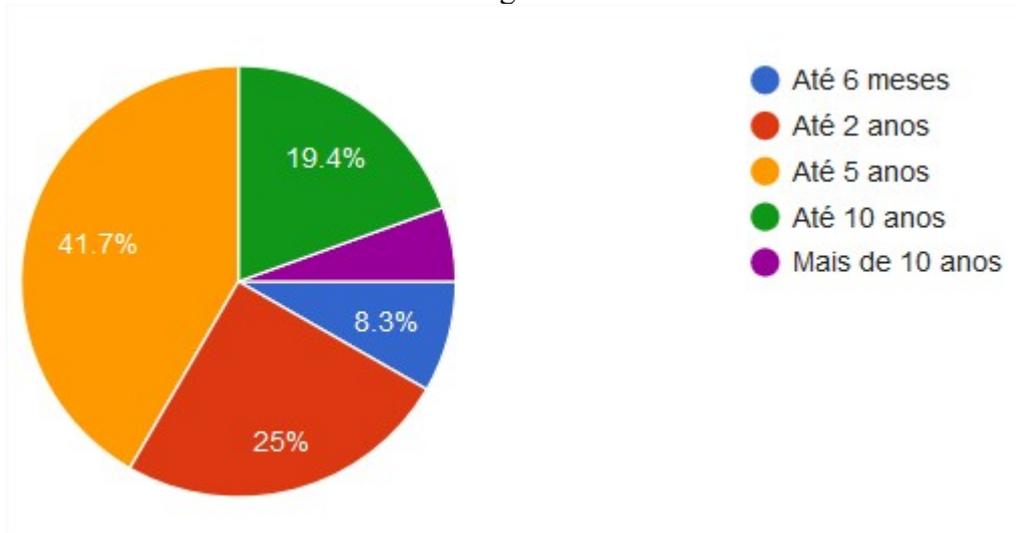


Fonte: Elaborado pelo autor

Os respondentes com mais de 10 anos de experiência em desenvolvimento de software formam o maior grupo, seguido pelos profissionais com até 5 anos de experiência. Estes dois grupos correspondem juntos 72,2% dos respondentes, mostrando uma polarização significativa de experiência em projetos de software da maioria dos respondentes.

Também foi coletada a informação do tempo de experiência em projetos que se utilizam de metodologias ágeis. A distribuição dos respondentes é descrita pela Figura 14.

Figura 14: Experiência em projetos de software que utilizam metodologias ágeis



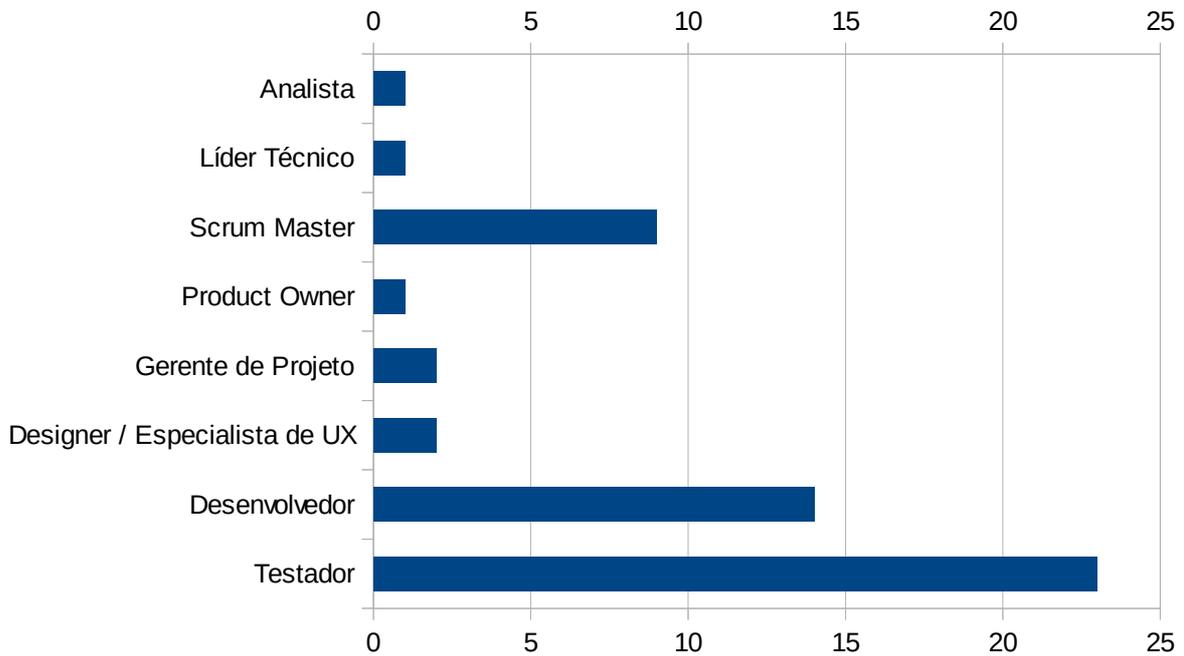
Fonte: Elaborado pelo autor

Neste quesito, 75% dos respondentes informaram ter até 5 anos de experiência em projetos que utilizam metodologias ágeis. Enquanto que apenas 5,6% dos respondentes comunicaram possuir mais de 10 anos experiência em projetos ágeis.

4.4.2 Papéis Desempenhados em Projetos de Software

A pergunta seguinte refere-se ao papel desempenhado pelo profissional nos projetos ágeis. Por não termos restringido a pesquisa a algum projeto específico e pela própria característica de times multifuncionais das metodologias ágeis, o respondente poderia selecionar mais de um papel. A Figura 15 mostra a distribuição dos respondentes neste quesito.

Figura 15: Distribuição de respondentes por papel em projetos ágeis



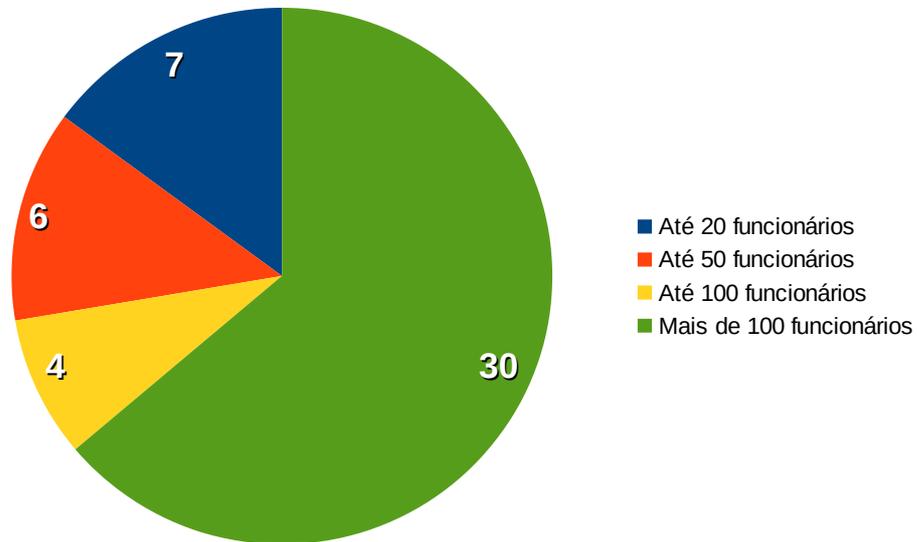
Fonte: Elaborado pelo autor

4.4.3 Características das Empresas

As próximas duas questões estavam relacionadas às características da empresas onde os projetos ágeis foram desenvolvidos. Novamente, devido a não restrição de resposta a um projeto ou empresa em específico, o respondente poderia selecionar mais de uma opção em cada uma das respostas da seção.

Primeiramente, foi perguntado o número de funcionários das empresas em que o profissional participou em projetos ágeis. A Figura 16 mostra a distribuição dos respondentes nesse quesito.

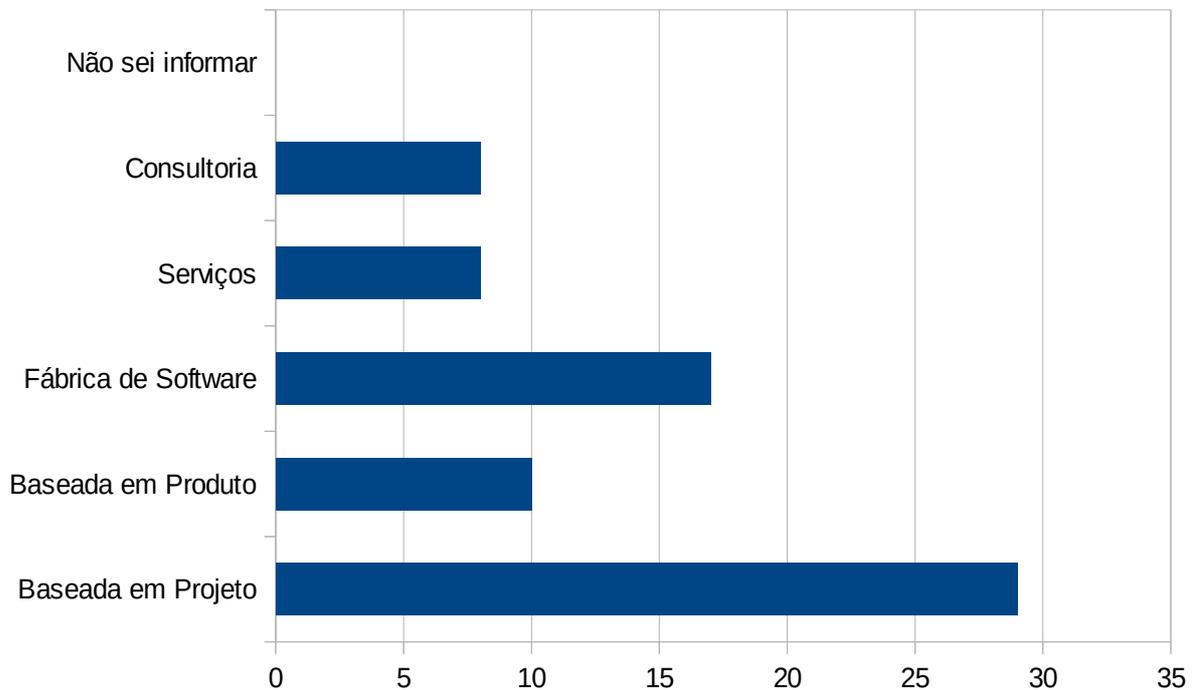
Figura 16: Número de funcionários nas empresas



Fonte: Elaborado pelo autor

Em seguida, foi indagado o tipo o contexto da área de desenvolvimento de software na empresa, de acordo com seu objetivo. A Figura 17 mostra a majoritariedade das empresas são baseadas em produto e fábricas de software.

Figura 17: Contexto de desenvolvimento das empresas



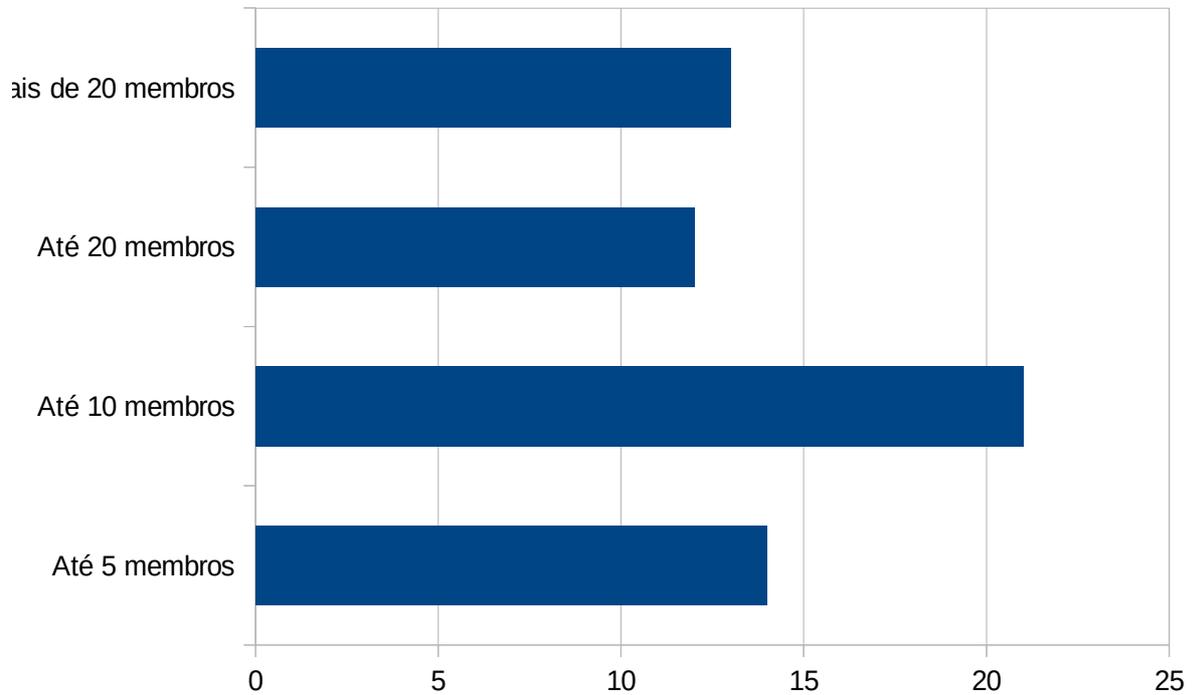
Fonte: Elaborado pelo autor

4.4.4 Características dos Projetos e Equipes de Desenvolvimento

Após levantar as características organizacionais, buscou-se entender o contexto dos projetos ágeis desenvolvidos, em especial da área de testes.

A primeira pergunta relativa aos projetos foi em relação ao número de membros das equipes. A Figura 18 mostra que 58,3% dos respondentes informaram participar de equipes com até 10 membros, tamanho ideal para equipes ágeis [51].

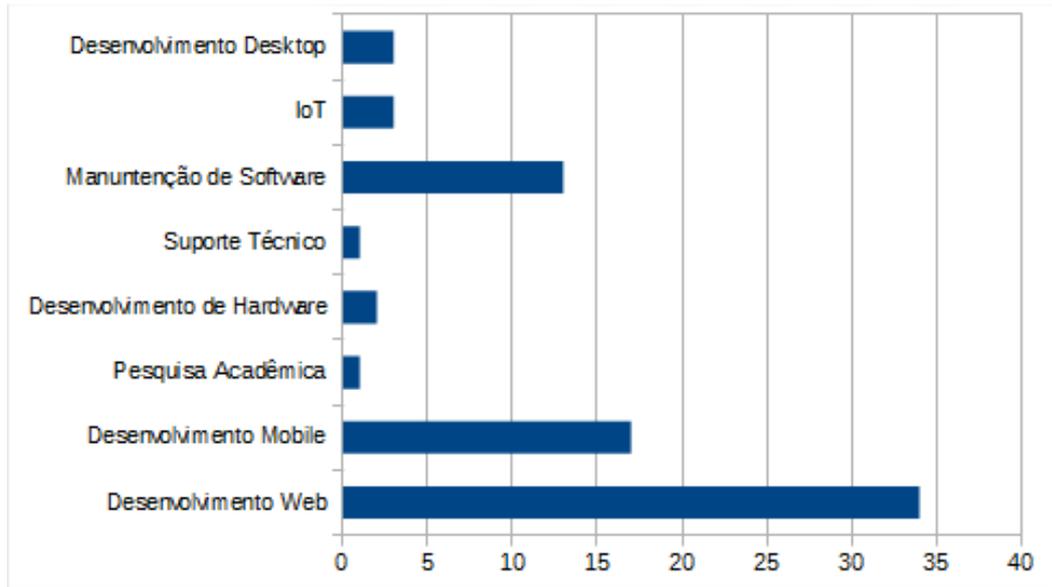
Figura 18: Tamanho das equipres ágeis



Fonte: Elaborado pelo autor

Em seguida, foi perguntado qual tipo de *software* os projetos desenvolviam. A Figura 19 mostra que as principais áreas de desenvolvimento web ou aplicativos móveis. Uma outra categoria representativa foram os projetos de manutenção de software..

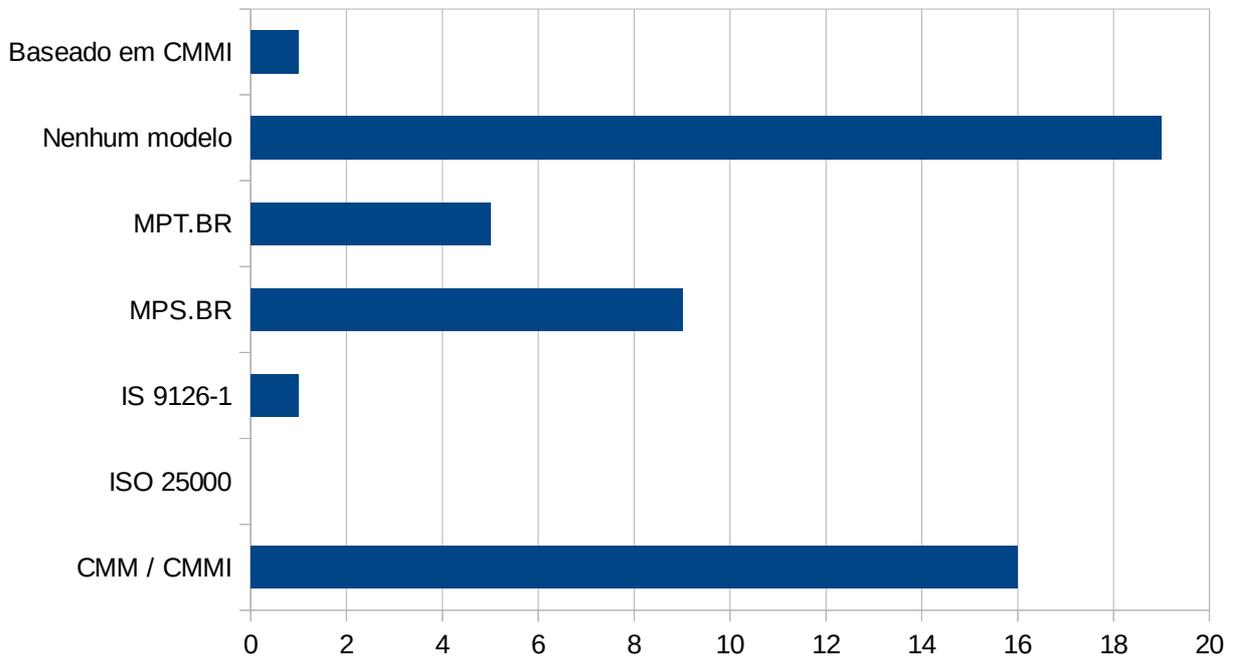
Figura 19: Área tecnológica dos projetos ágeis



Fonte: Elaborado pelo autor

A terceira pergunta sobre as características dos projetos buscava obter informações sobre possíveis modelos de maturidade de software implementados nos projetos. A Figura 20 mostra que 52,8% dos respondentes trabalharam em projetos não seguiam nenhum modelo de maturidade estabelecido na indústria. Dos projetos que citaram o uso de algum modelo, 53% informaram implementar o modelo CMM/CMMI; o segundo modelo mais citado foi o MPS.BR.

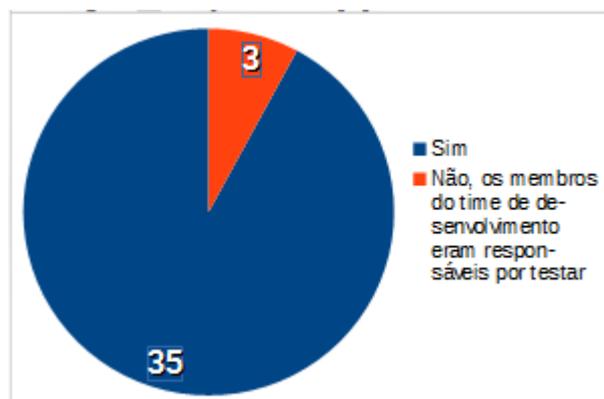
Figura 20: Modelo de maturidade de projetos ágeis



Fonte: Elaborado pelo autor

As três perguntas seguintes nesta seção indagaram o respondente sobre características dos profissionais de testes nos projetos ágeis. A Figura 21 mostra que 97% dos respondentes trabalharam em projetos com profissionais de teste dedicados. Apenas 3 respondentes informaram trabalhar em projetos em que os próprios desenvolvedores eram responsáveis pela garantia de qualidade do produto.

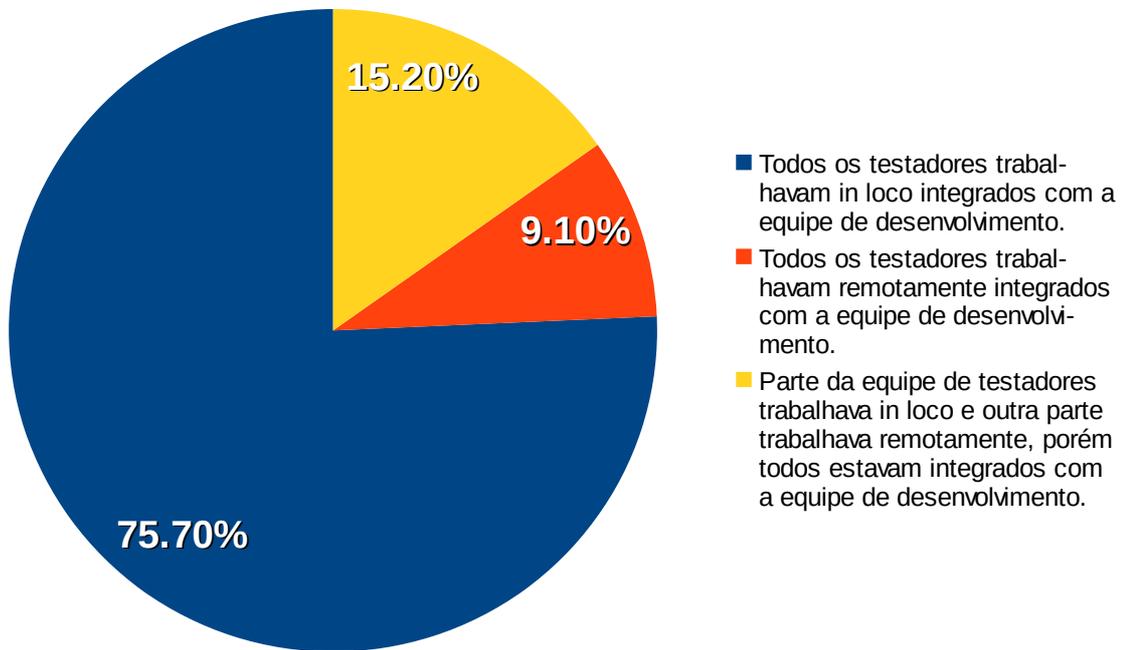
Figura 21: Projetos com equipe de teste dedicada



Fonte: Elaborado pelo autor

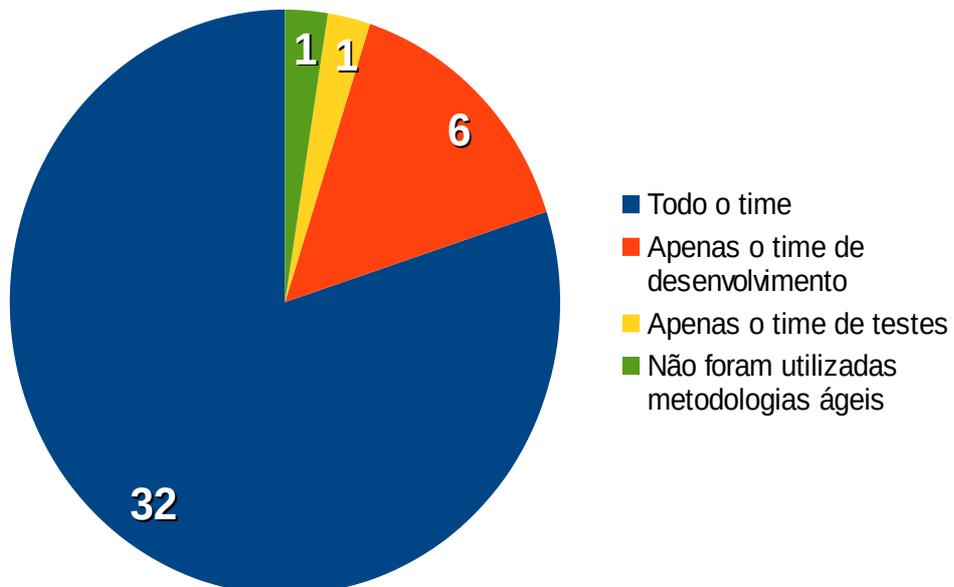
Em seguida, os respondentes que trabalharam em projetos com equipes de teste dedicadas informaram sobre a integração das equipes de desenvolvimento e testes, relacionadas à geolocalização de seus membros. Como mostra a Figura 22, 75,8% dos respondentes informaram que as equipes de desenvolvimento e testes trabalharam dentro do mesmo espaço físico, o que incentiva a comunicação entre os membros do time, levando a soluções mais rápidas de problemas [13].

Figura 22: Distribuição geográfica das equipes de teste



Fonte: Elaborado pelo autor

Figura 23: Áreas do projeto que utilizaram metodologias ágeis



Fonte: Elaborado pelo autor

A última pergunta da seção indagou os respondentes sobre a utilização de metodologias ágeis pelas equipes de teste e desenvolvimento. Cerca de 17% dos respondentes informaram que as equipes de teste e desenvolvimento implementavam metodologias diferentes, dividindo-se entre processos ágeis e não-ágeis dentro do mesmo projeto, como mostra a Figura 23.

4.5 Práticas de Testes Ágeis

A terceira seção do questionário é denominada “Os Quadrantes de Testes Ágeis”, composta de 20 perguntas. O objetivo desta seção é mapear quais técnicas de testes ágeis foram utilizadas pelos respondentes, seus contextos de uso e valor agregado ao projeto.

Primeiramente, o modelo de Quadrantes de Testes Ágeis foi brevemente descrito, afim de que o respondente possa entender melhor o contexto de cada sub-seção. Em seguida, 4 sub-seções abordaram isoladamente cada quadrante, levantam as seguintes informações:

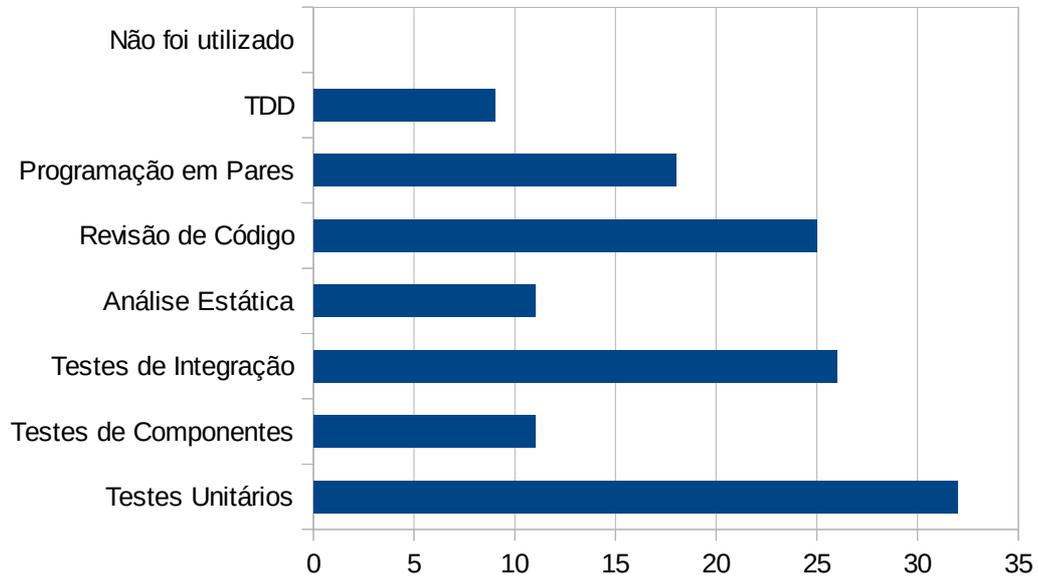
- Quais técnicas relativas ao quadrante foram utilizadas?
- Qual o papel dos profissionais que exercitaram esses testes?
- Com que frequência esse tipo de teste foi realizado?
- Em que etapa(s) do projeto esse tipo de teste foi realizado?
- Qual contribuição esse tipo de teste trouxe ao projeto?

O restante dessa seção sumariza as respostas para cada Quadrante de Testes Ágeis.

4.5.1 Práticas do Primeiro Quadrante de Testes Ágeis

A Figura 24 informa as técnicas relativas ao Primeiro Quadrante de Testes Ágeis. Observa-se que as técnicas mais utilizadas foram Testes Unitários (88,9%), Testes de Integração (72,2%) e Revisão de Código (69,4%).

Figura 24: Distribuição das técnicas relativas ao Primeiro Quadrante de Testes Ágeis

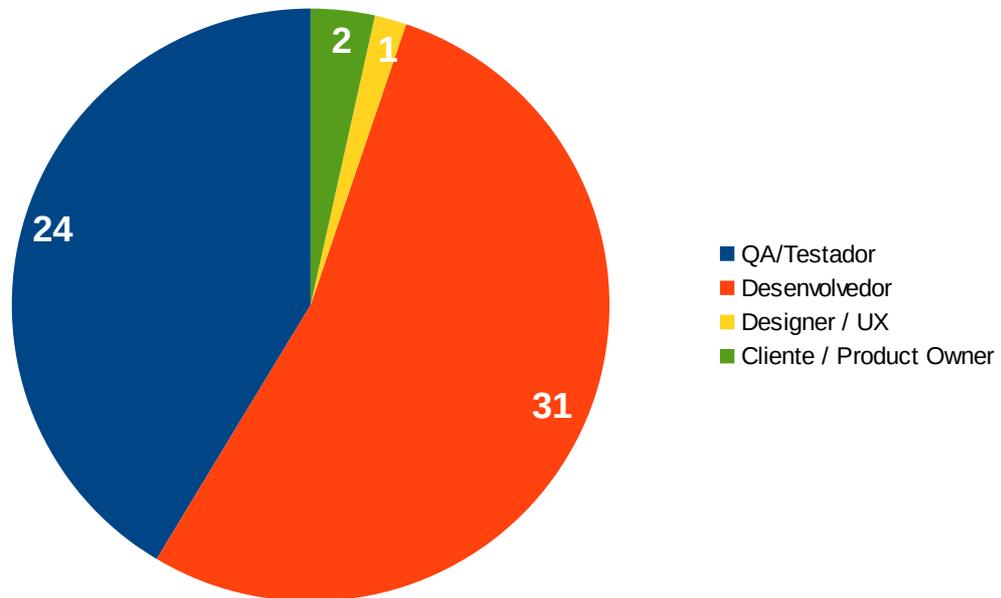


Fonte: Elaborado pelo autor

O fato de apenas 25% dos respondentes terem informado a utilização de *Test-Driven Development* pode indicar que as técnicas de testes unitários, de integração e de componentes eram utilizadas com o objetivo de verificar possíveis regressões de funcionalidade. Nesse contexto, esses testes não são trazem valor ótimo ao projeto, pois não são utilizados para dirigir o desenvolvimento de código mais simples e testável [40].

Desenvolvedores e Testadores foram majoritariamente os responsáveis pelo desenvolvimento e execução desse tipo de teste, como mostra a Figura 25. Apenas três respondentes informaram que profissionais mais ligados ao negócio (Designers e Cliente/Product Owner) envolveram-se nos testes desse quadrante.

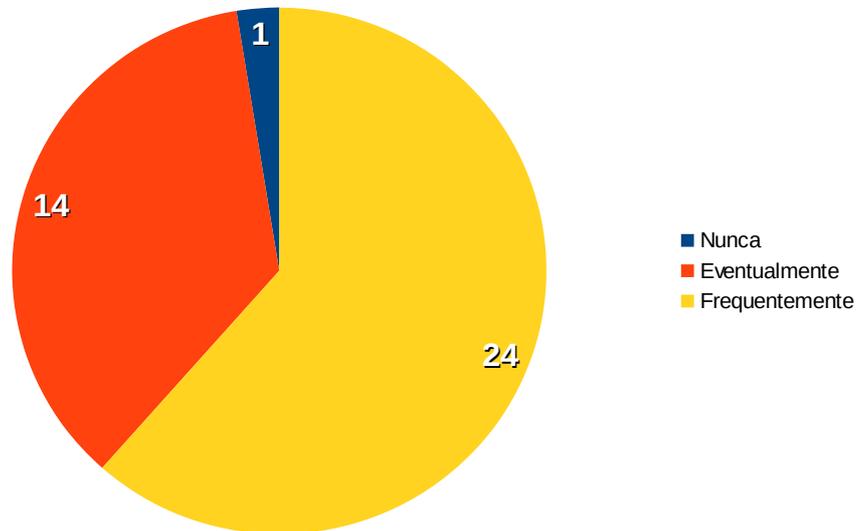
Figura 25: Distribuição dos papéis relativos ao Primeiro Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

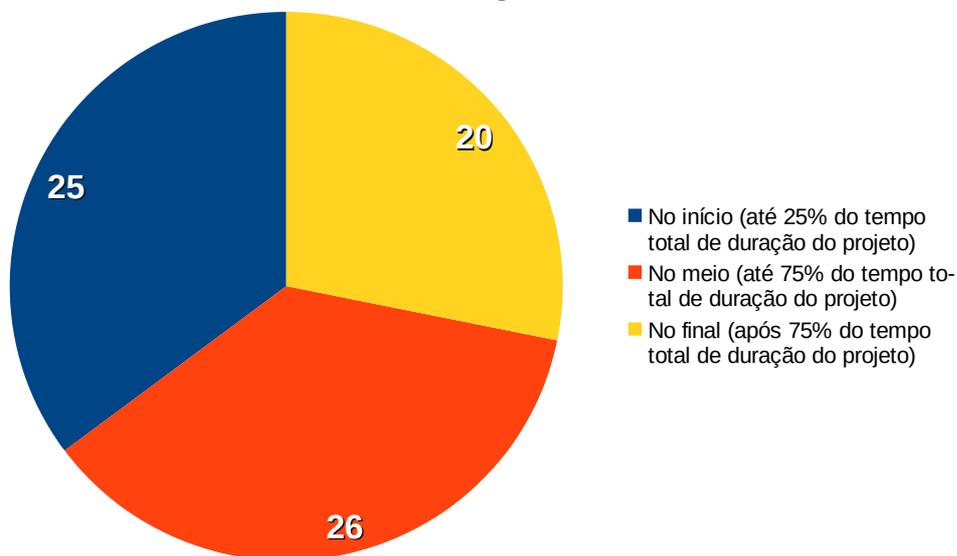
Os respondentes informaram que, comumente, as técnicas classificadas no Primeiro Quadrante de Testes Ágeis foram utilizadas frequentemente durante toda a vida dos projetos ágeis, como mostram as Figura 26 e 27.

Figura 26: Frequência de utilização das técnicas relativas ao Primeiro Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

Figura 27: Etapa de projeto de uso das técnicas relativas ao Primeiro Quadrante de Testes Ágeis



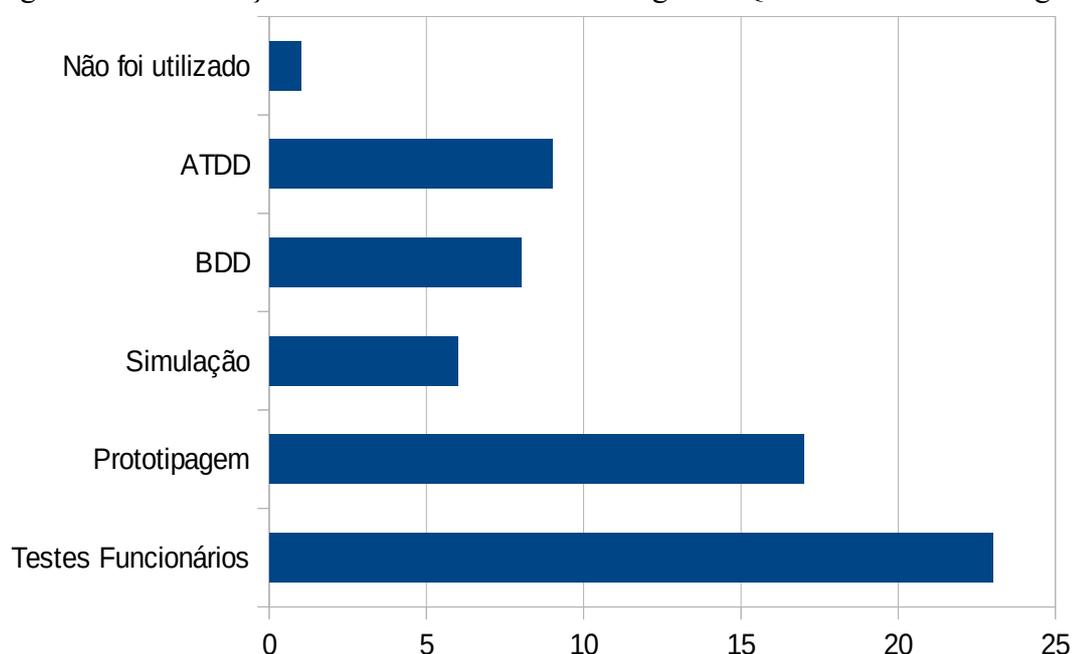
Fonte: Elaborado pelo autor

Em geral, na visão dos respondentes, os testes do Primeiro Quadrante de Testes Ágeis foram benéficos aos projetos. Foi informado que os testes do quadrante Q1 permitiram “reduzir a quantidade de bugs menor” que escapavam da fase de desenvolvimento, assim, “diminuindo o tempo/custo gasto nas fases seguintes do projeto”.

4.5.2 Práticas do Segundo Quadrante de Testes Ágeis

A Figura 28 informa as técnicas relativas ao Segundo Quadrante de Testes Ágeis. Observa-se que as técnicas mais utilizadas referiam-se a verificação automatizada das funcionalidades implementadas (Testes Funcionais Automatizados) e a validação do entendimento dos requisitos (Prototipagem e Simulação). Enquanto que metodologias de desenvolvimento dirigidas por esses tipos de teste (BDD e ATDD) foram menos citadas.

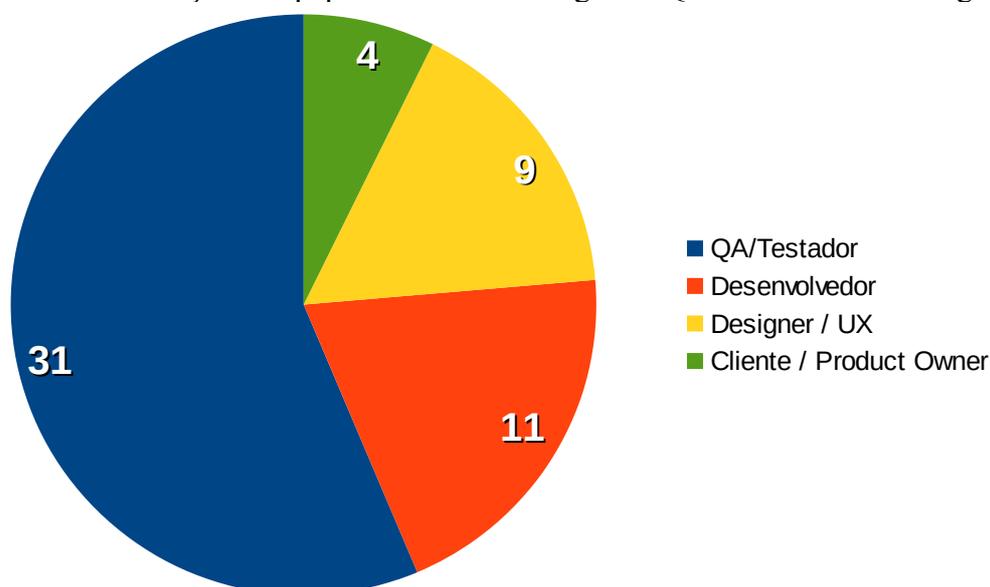
Figura 28: Distribuição das técnicas relativas ao Segundo Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

Os Testadores foram os principais executores desses testes, como mostra a Figura 29. No espaço para comentários dessa seção, foi informado que os Testadores eram responsáveis pelos testes de validação de requisitos devido a capacidade inata desse tipo de profissional de identificar *edge cases* e, assim, ajudar o *Product Owner* a definir mais precisamente o comportamento esperado dos sistemas.

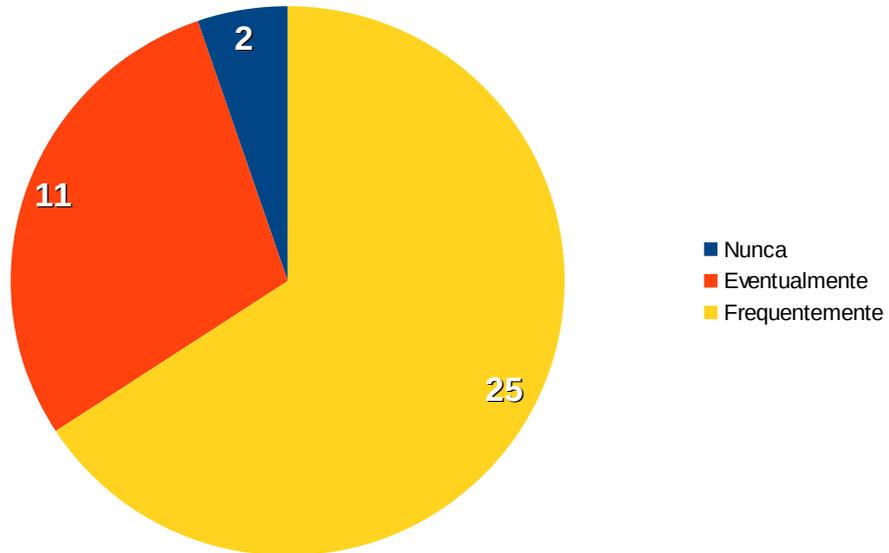
Figura 29: Distribuição dos papéis relativos ao Segundo Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

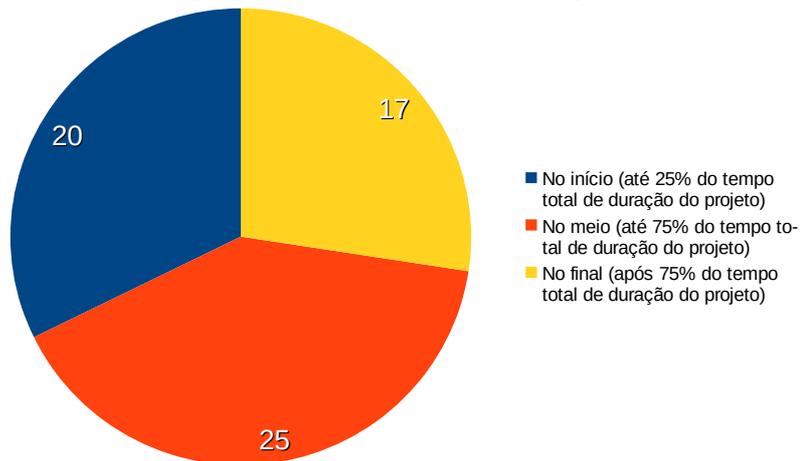
Os respondentes informaram que, comumente, as técnicas classificadas no Segundo Quadrante de Testes Ágeis foram utilizadas frequentemente durante toda a vida dos projetos ágeis, como mostram as Figuras 30 e 31.

Figura 30: Frequência de utilização das técnicas relativas ao Segundo Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

Figura 31: Etapa de projeto de uso das técnicas relativas ao Segundo Quadrante de Testes Ágeis



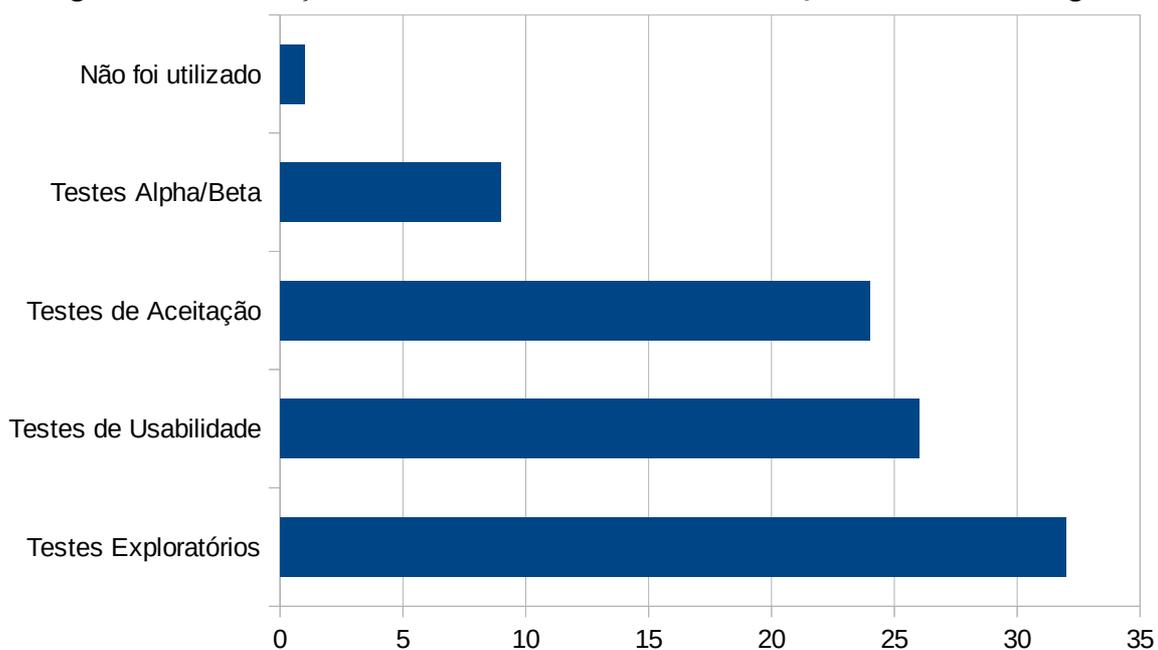
Fonte: Elaborado pelo autor

Em geral, na visão dos respondentes, os testes do quadrante Q2 foram benéficos aos projetos. Foi informado que os testes do quadrante Q2 permitiram diminuir o re-trabalho em etapas posteriores ao desenvolvimento e validar a viabilidade de um projeto ou funcionalidade antes de sua implementação.

4.5.3 Práticas do Terceiro Quadrante de Testes Ágeis

A Figura 32 informa as técnicas relativas ao Terceiro Quadrante de Testes Ágeis. Observa-se que as técnicas mais utilizadas foram os Testes Exploratórios (88,9%), Testes de Usabilidade (72,2%) e Testes de Aceitação (66,7%).

Figura 32: Distribuição das técnicas relativas ao Terceiro Quadrante de Teste Ágeis

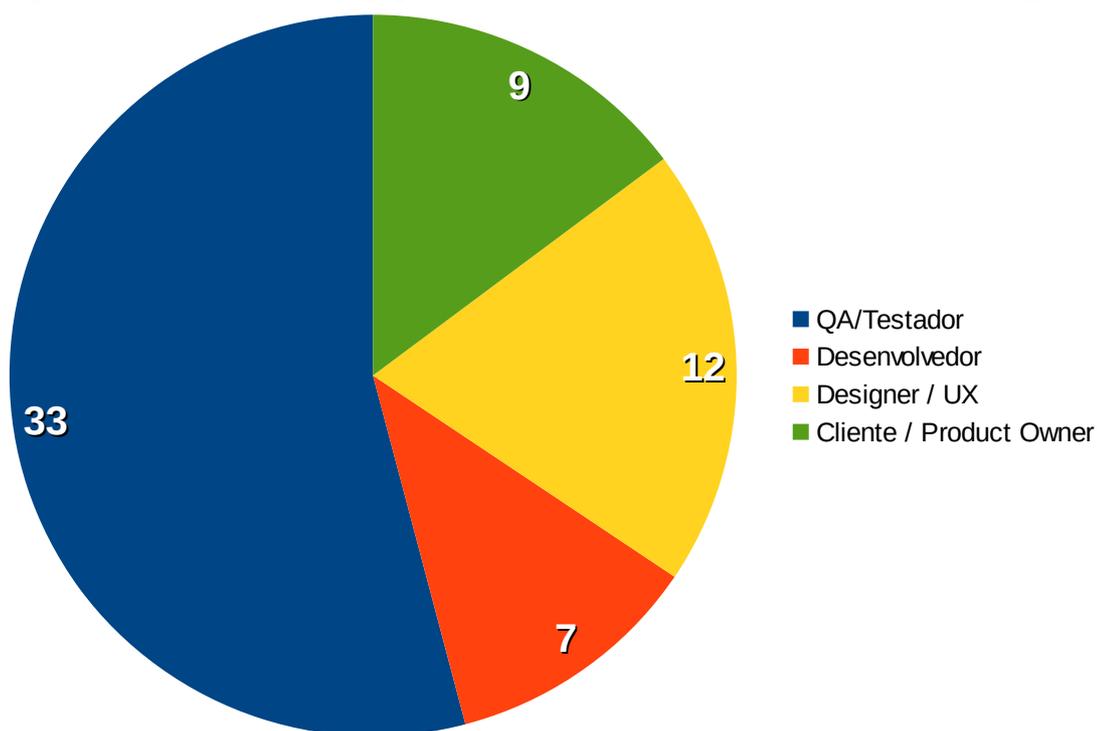


Fonte: Elaborado pelo autor

Apesar das três técnicas mais citadas terem o mesmo objetivo, validar o sistema ou funcionalidade, elas investigam o produto sob perspectivas diferentes. Testes exploratórios são realizados por testadores, buscando encontrar problemas que até usuários comuns dificilmente perceberiam, como inconsistências com outros sistemas. Testes de usabilidade buscam validar se a aplicação pode ser usada pelo usuário-alvo de forma eficiente. Enquanto testes de aceitação apresentam o sistema implementado ao *stakeholder*, com o objetivo de validar se a sua necessidade foi plenamente atendida.

Os Testadores foram os principais executores desses testes, seguidos pelos profissionais de Designer e Experiência do Usuário, como mostra a Figura 33.

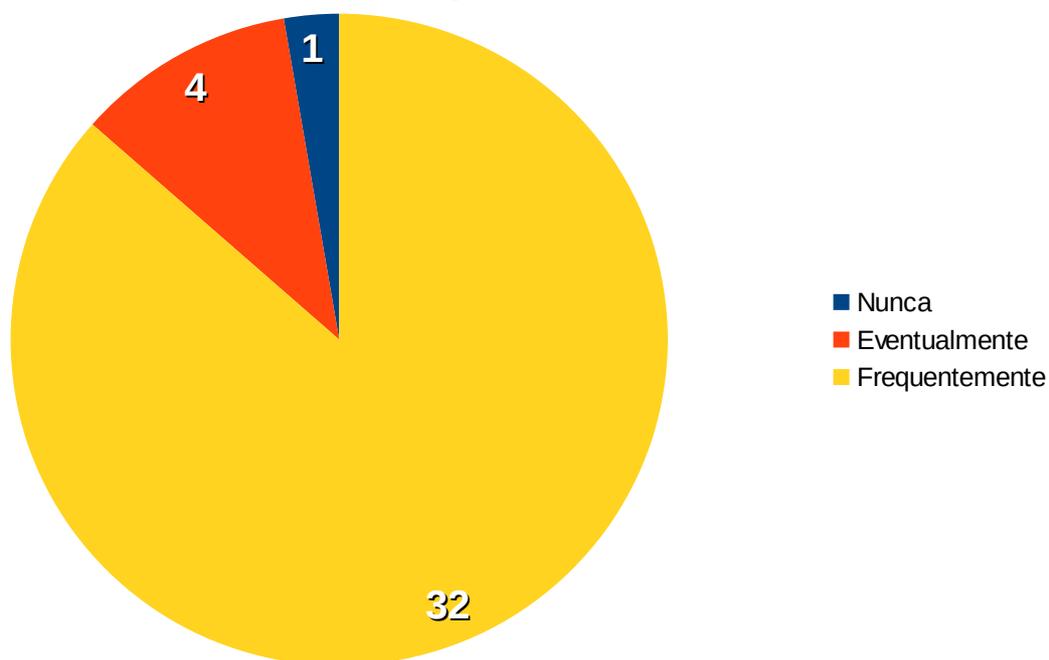
Figura 33: Distribuição dos papéis relativos ao Terceiro Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

Os respondentes informaram que, comumente, as técnicas classificadas no quadrante Q3 foram utilizadas frequentemente durante as iterações (como mostra a Figura 34), porém, o início da aplicação deles deu-se em etapas tardias dos projetos, como mostra a Figura 35.

Figura 34: Frequência de utilização das técnicas relativas ao Terceiro Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

Figura 35: Etapa de projeto de uso das técnicas relativas ao Terceiro Quadrante de Testes Ágeis



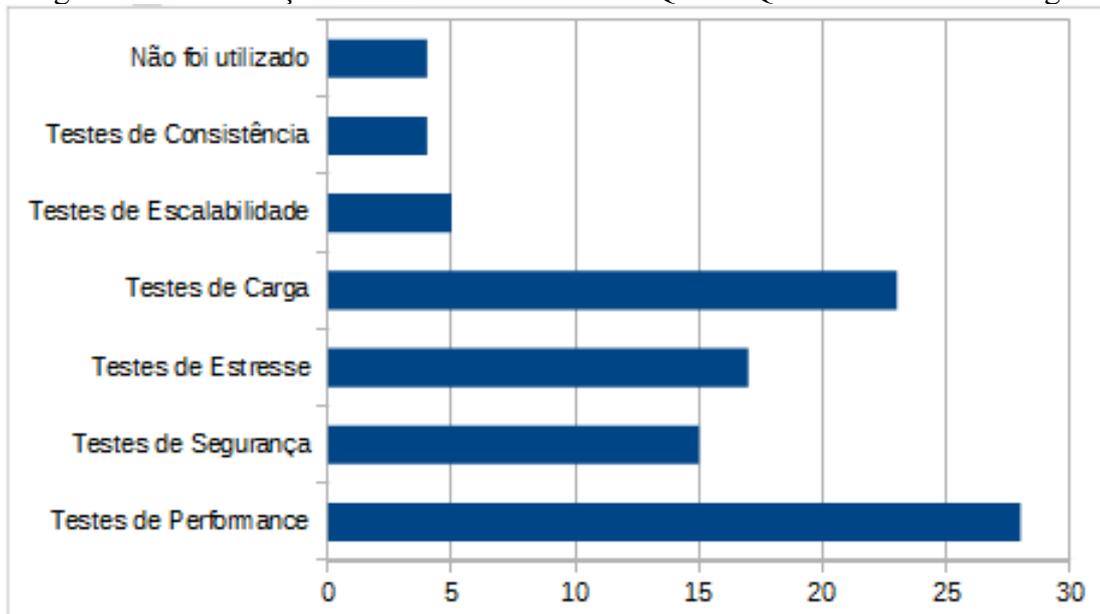
Fonte: Elaborado pelo autor

Esse fenômeno pode ser decorrente do processo de aprendizado de equipe em projetos de desenvolvimento ágil. Devido as curtas iterações e *feedback* rápido, equipes ágeis tendem a descobrir, durante o projeto, melhores maneiras de trabalhar, alterando seus processos e técnicas com o objetivo de aumentar sua produtividade.

4.5.4 Práticas do Quarto Quadrante de Testes Ágeis

A Figura 36 informa as técnicas relativas ao Quarto Quadrante de Testes Ágeis. Observa-se que as técnicas mais utilizadas foram os Testes Performance (77,8%) e Testes de Carga (77,8%).

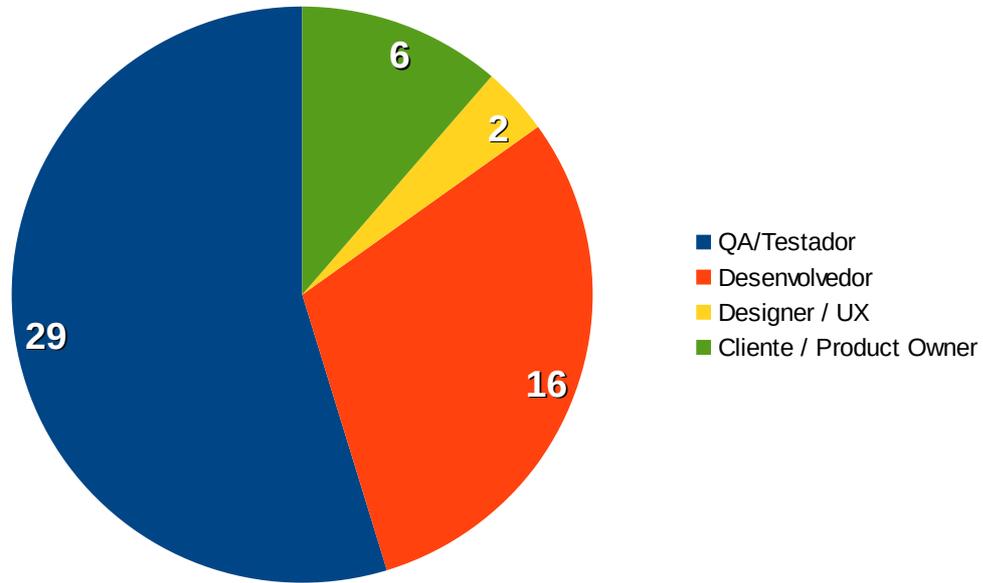
Figura 36: Distribuição das técnicas relativas ao Quarto Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

Os Testadores e Desenvolvedores foram os principais executores desses testes, como mostra a Figura 37. Crispin e Gregory [14] recomendam que as atividades relacionadas a estes testes sejam executadas por Desenvolvedores e Testadores. Devido à natureza técnica desta categoria de testes, os Desenvolvedores são essenciais para a criação de uma solução robusta, enquanto que os Testadores são capazes de explorar os sistemas de forma a descobrir erros.

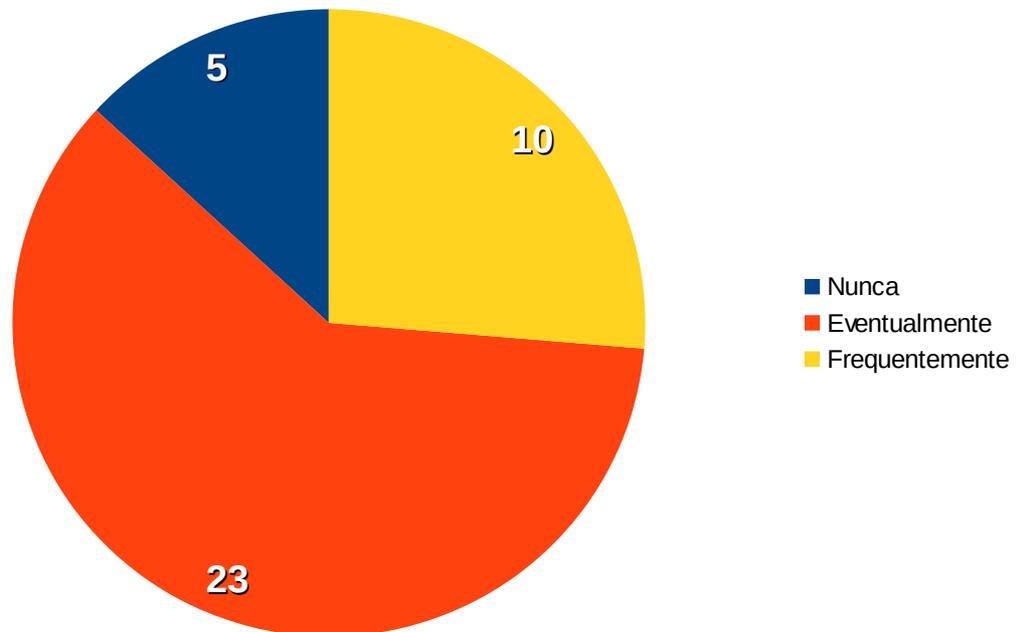
Figura 37: Distribuição dos papéis relativos ao Quarto Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

A Figura 38 mostra que, diferentemente das outras categorias, os testes classificados no quadrante Q4 foram realizadas de forma eventual. Um respondente justificou essa situação ao dizer que “esses tipos de testes são aplicados sob demanda, dependendo dos requisitos não funcionais definidos. Depende da natureza do projeto. Por exemplo, segurança é importante se for um aplicativo aberto para o público externo.”

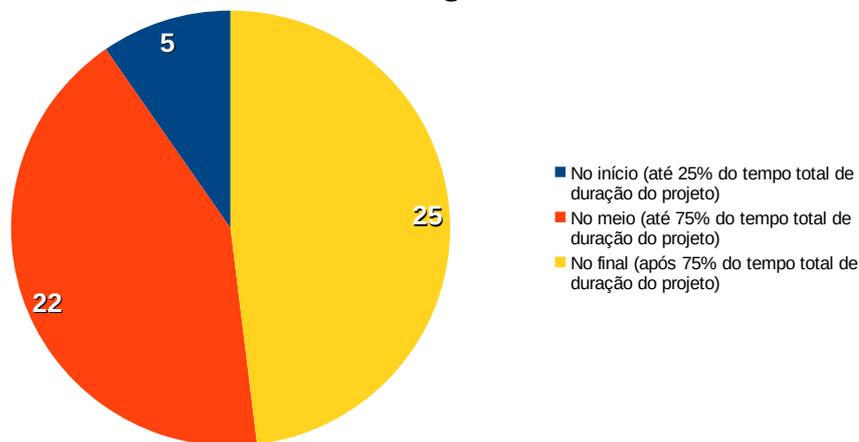
Figura 38: Frequência de utilização das técnicas relativas ao Quarto Quadrante de Testes Ágeis



Fonte: Elaborado pelo autor

Também foi observado, como mostra a Figura 39, que esses testes são geralmente aplicados em etapas tardias do projeto. Como mostrado na Seção 2.4.2, esses testes podem revelar problemas fundamentais em decisões arquiteturais, em quesitos como escalabilidade e estabilidade, que podem escapar das outras categorias de testes. Um respondente enfatizou esse aspecto ao falar que “erros encontrados nessa fase ajudam a melhorar a arquitetura do projeto o mais cedo possível ou tomar ações relativas a uma melhor abordagem para viabilizar a solução dos problemas.”

Figura 39: Etapa de projetos de uso das técnicas relativas ao Quarto Quadrante de Testes Ágeis



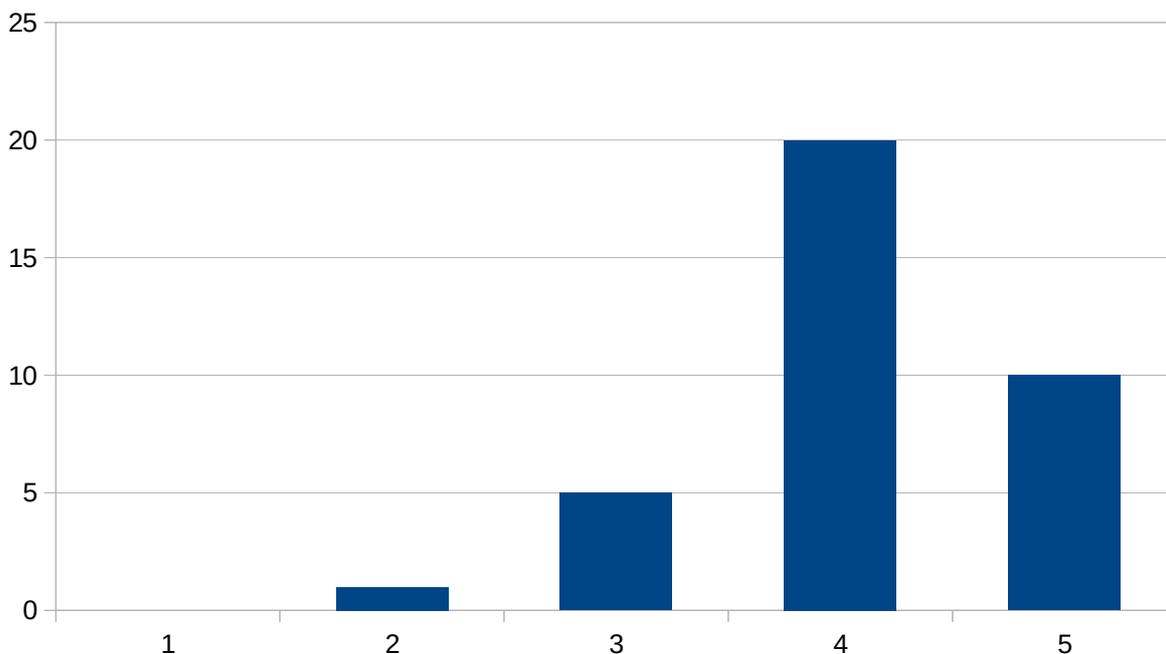
Fonte: Elaborado pelo autor

4.6 Percepções Sobre Testes Ágeis

A quarta seção do questionário é denominada “Percepções sobre Testes Ágeis”, composta de 4 perguntas. O objetivo desta seção é entender os benefícios e desafios nos os processos de teste em projetos ágeis.

Primeiramente, foi pedido ao respondente para comparar a qualidade dos produtos desenvolvidos utilizando metodologias ágeis nas atividades de teste com produtos desenvolvidos utilizando metodologias tradicionais. Utilizou-se uma escala de 1 a 5, sendo 1 equivalente a “Muito pior” e 5 equivalente a “Muito melhor”. A Figura 40 demonstra que a opinião dos respondentes é que as técnicas de testes ágeis estão relacionadas a produtos de maior qualidade.

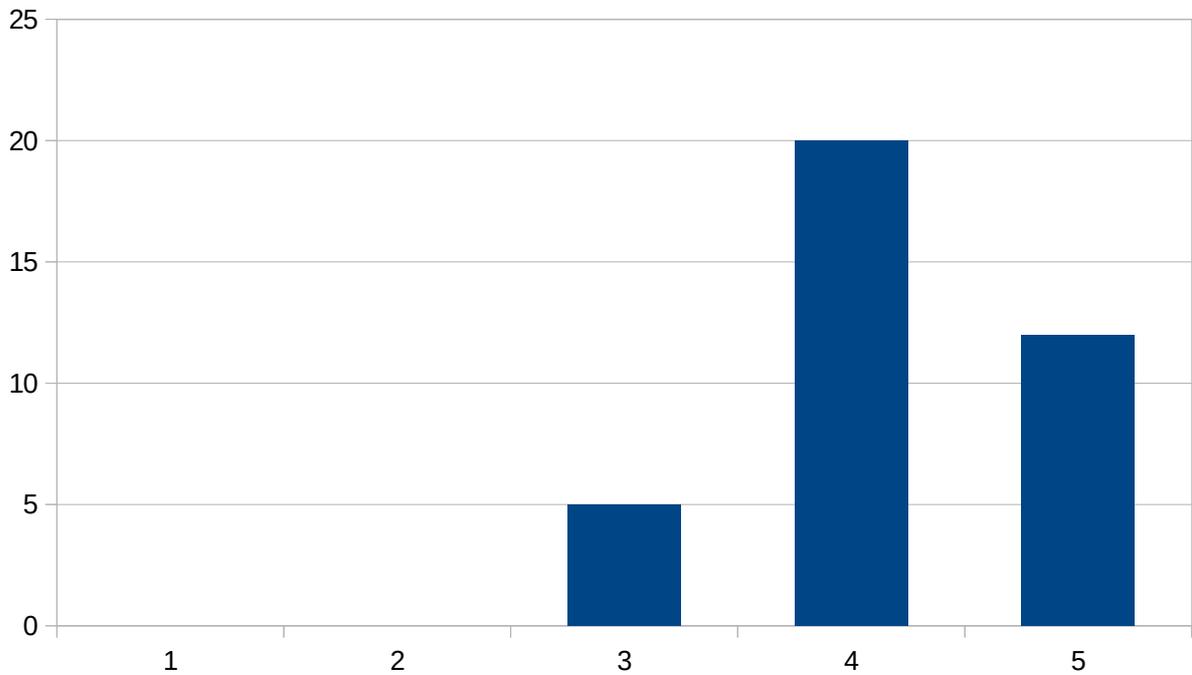
Figura 40: Qualidade dos produtos de projetos ágeis em comparação com não-ágeis



Fonte: Elaborado pelo autor

Em seguida, pediu-se ao respondente para comparar o nível de engajamento das equipes que utilizaram-se de metodologias ágeis nas atividades de teste com àquelas que utilizaram metodologias tradicionais. Utilizou-se uma escala de 1 a 5, sendo 1 equivalente a “Muito menor” e 5 equivalente a “Muito maior”. A Figura 41 demonstra que a opinião dos respondentes é que as técnicas de testes ágeis estão relacionadas com maior engajamento dos membros das equipes nos projetos.

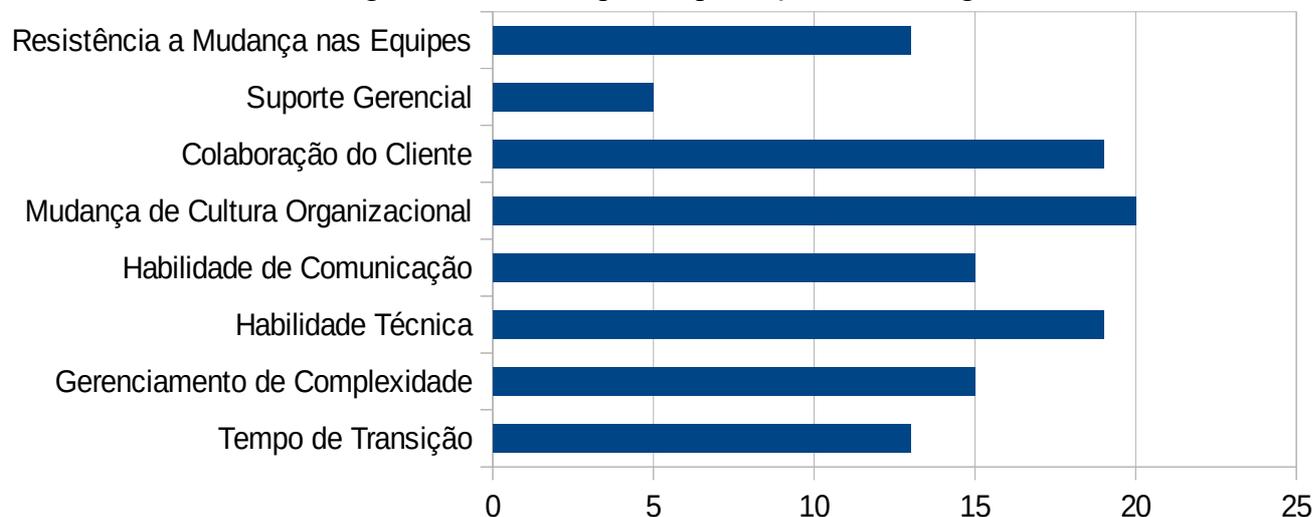
Figura 41: Nível de engajamento das equipes de projetos ágeis em comparação com não-ágeis



Fonte: Elaborado pelo autor

Afim de identificar problemas comuns com equipes que utilizam metodologias ágeis nos processos de teste, foi pedido para os respondentes listarem os principais desafios para implantação e execução de testes ágeis. A Figura 42 mostra que alguns dos principais desafios estão relacionados com cultura organizacional e relacionamento com *stakeholders* externos, corroborando com com outros trabalhos encontrados na literatura [2][3].

Figura 42: Desafios para implantação de testes ágeis



Fonte: Elaborado pelo autor

4.7 Considerações Finais

Através da análise dos dados coletados, pode-se concluir que as empresas do Porto Digital do Recife conseguem perceber o valor agregado aos produtos através das práticas de testes ágeis. Corroborando com os achados do mapeamento sistemático apresentado no Capítulo 3, a aplicação de técnicas de testes ágeis permitiram aos times aumentar a qualidade interna e externa dos produtos. Além disso, técnicas do quadrante Q2, também mostram-se úteis para diminuir a quantidade de re-trabalho devido a problemas na fase de levantamento de requisitos.

Dificuldades para a implantação dessas práticas são semelhantes às encontradas em outros centros de desenvolvimento, estando ligadas majoritariamente à capacidade técnica dos profissionais e mudanças de cultura organizacional. O baixo número de

respondentes que informaram utilizar TDD (25%), Análise Estática (30,6%) e Programação em Pares (50%) pode indicar uma oportunidade de melhoria no sentido que estas técnicas promovem rápido *feedback* sobre qualidade interna e externa do *software*, como indicam estudos selecionados no mapeamento sistemático do Capítulo 3 [36][37]. Ademais, os respondentes informaram que as validações trazidas pelos testes do quarto quadrante, em geral, são postergadas para as fases finais dos projetos. Segundo, Lucena e Tizzei [38], apesar dessa prática ser comum na indústria brasileira, esse adiamento aumenta o custo de realizar as mudanças necessárias para aumentar a qualidade dos produtos.

O capítulo seguinte encerra a pesquisa, apresentando suas conclusões, contribuições, limitações e propostas de trabalhos futuros.

5 Conclusão

Neste capítulo, serão apresentados as considerações finais, as principais contribuições e os possíveis trabalhos futuros.

5.1 Considerações Finais

Este trabalho de graduação buscou entender a prática de teste de software em projetos ágeis no Brasil e no Porto Digital do Recife. Para tanto, foi realizado um mapeamento sistemático da pesquisa realizada no Brasil e um questionário foi respondido por profissionais da indústria que atuam em projetos de desenvolvimento ágil no Porto Digital do Recife.

Utilizando o modelo de Quadrantes de Testes Ágeis, foram identificados, na literatura brasileira, estudos que mostram os benefícios e desafios das práticas de testes ágeis. Testes unitários e automação de testes funcionais capacitaram os desenvolvedores a lidar com mudanças frequentes nos sistemas, sem diminuir a qualidade do código. Inspeções de usabilidade mitigaram erros na etapa de levantamento de requisitos, onde o custo de mudança é o menor no ciclo de desenvolvimento de software. Testes de usabilidade e testes exploratórios garantiram que o produto desenvolvido atendia todas as necessidades dos usuários. Testes de performance e segurança mostram-se ferramentas fundamentais para garantir o uso estável do sistema em ambiente de produção. Porém, também foi identificado que comumente esses testes são postergados para as fases finais do projeto, quando as mudanças arquiteturais nos sistemas tem alto custo e risco.

A pesquisa com profissionais do Porto Digital identificou as principais técnicas de testes em projetos ágeis utilizadas nos projetos, e seus contextos de aplicação. Ademais, foram levantadas os principais benefícios e desafios na prática de testes ágeis. Pode ser observado, corroborando com os achados do mapeamento sistemático, que as práticas de testes são exercitadas em todas as etapas do ciclo de desenvolvimento de software, trazendo diversos benefícios para a qualidade dos produtos, porém, falta de capacidade técnica e mudança de cultura organizacional mostram-se dificuldades de importantes para implantação plena das metodologias ágeis de teste de software.

5.2 Limitações

Com relação ao mapeamento sistemático da literatura delineado no Capítulo 3, aproximadamente 12% dos trabalhos selecionados no processo de *snowballing* descrito na Seção 3.2.2 não puderam ser analisados completamente, pois não foi possível adquirí-los na rede da UFPE. Possivelmente alguns trabalhos relevantes não foram incluídos na análise.

No tocante da pesquisa de campo descrita no Capítulo 4, dado que a amostra de respondentes é não probabilística, os resultados não podem ser generalizados para toda a população do Porto Digital do Recife. Ademais, alguns *clusters* de respondentes, por exemplo funcionários de empresas com mais de 100 funcionários ou que atuaram no papel de Testador, podem ter enviesado os resultados das perguntas subjetivas para contextos específicos.

5.3 Trabalhos Futuros

Com relação ao mapeamento sistemático, é sugerido como trabalho futuro a validação estatística da efetividade da estratégia de *snowballing* utilizada, em comparação com a técnica apresentada por Wohlin [35].

Devido a restrições de tempo e custo, a pesquisa de campo limitou-se quase que exclusivamente a perguntas objetivas. Entretanto, a utilização de perguntas subjetivas enriquece o entendimento de um contexto social. Portanto, sugere-se que próximas pesquisas abram mais espaço para os respondentes dissertarem sobre cada aspecto questionado. Ademais, este estudo buscou a obter uma visão do estado presente da prática de testes ágeis no Porto Digital. Observar como as equipes desenvolvem as habilidades e implementam as metodologias ao longo do tempo dos projetos é uma sugestão de trabalho futuro.

6 Referências

- [1] SOMMERVILLE, Ian. Software Engineering. Addison Wesley, 2015.
- [2] SAUCE LABS. Testing Trends in 2017: A Survey of Software Professionals. Disponível em: <<https://saucelabs.com/news/sauce-labs-releases-third-annual-state-of-testing-surveyresults>>. Acesso em: 29 jun. 2017.
- [3] MELO, Claudia O. et al. The evolution of agile software development in Brazil. Journal of the Brazilian Computer Society, v. 19, n. 4, p. 523, 2013.
- [4] MYERS, Glenford J.; SANDLER, Corey; BADGETT, Tom. The art of software testing. John Wiley & Sons, 2011.
- [5] RODRIGUES SOARES, Elifrancis. Adaptação do processo de desenvolvimento de software para análise de cobertura de código. 2007.
- [6] BULLOCK, James. Calculating the Value of Testing, 2000. Disponível em: <<https://www.stickyminds.com/better-softwaremagazine/calculating-value-testing>>. Acesso em 29 de jun. 2017
- [7] DAHL, Ole-Johan; DIJKSTRA, Edsger Wybe; HOARE, Charles Antony Richard. Structured programming. Academic Press Ltd., 1972.
- [8] VIANA, Virginia Maria Araújo. Um método para seleção de testes de regressão para automação. 2006.
- [9] ZIV, Hadar; RICHARDSON, Debra; KLÖSCH, Rene. The uncertainty principle in software engineering. In: submitted to Proceedings of the 19th International Conference on Software Engineering (ICSE'97). 1997.
- [10] SCHWABER, Ken; BEEDLE, Mike. Agile software development with Scrum. Upper Saddle River: Prentice Hall, 2002.
- [11] WILLIAMS, Laurie; COCKBURN, Alistair. Guest Editors' Introduction: Agile Software Development: It's about Feedback and Change. Computer, v. 36, n. 6, p. 39-43, 2003.

- [12] FOWLER, Martin; HIGHSMITH, Jim. The agile manifesto. *Software Development*, v. 9, n. 8, p. 28-35, 2001.
- [13] CRISPIN, Lisa; GREGORY, Janet. *Agile testing: A practical guide for testers and agile teams*. Pearson Education, 2009.
- [14] MESZAROS, Gerard. *xUnit test patterns: Refactoring test code*. Pearson Education, 2007.
- [15] BOEHM, Barry W. Understanding and controlling software costs. *Journal of Parametrics*, v. 8, n. 1, p. 32-68, 1988.
- [16] PATTON, Ron. *Software testing*. Pearson Education India, 2006.
- [17] BECK, Kent. *Test-driven development: by example*. Addison-Wesley Professional, 2003.
- [18] VOGELSANG, Andreas; FEHNKER, Ansgar; HUUCK, Ralf; REIF, Wolfgang. Software metrics in static program analysis. *Formal Methods and Software Engineering (ICFEM 2010)*. 2010. p. 485-500
- [19] DEL FRATE, Fabio; GARG, Parul; MATHUR, Aditya; PASQUINI, Laura. On the correlation between code coverage and software reliability. In: *Software Reliability Engineering. Proceedings, Sixth International Symposium*. 1995. p. 124-132.
- [20] RTCA (FIRM). SC 167. *Software considerations in Airborne Systems and equipment certification*. RTCA, Incorporated, 1992.
- [21] ISO. ISO:26262-6:2011 *Road vehicles – functional safety - part 6: Product development at the software level*, 2011
- [22] WILLIAMS, Laurie et al. Strengthening the case for pair programming. *IEEE software*, v. 17, n. 4, p. 19-25, 2000.
- [23] COCKBURN, Alistair; WILLIAMS, Laurie. The costs and benefits of pair programming. *Extreme programming examined*, p. 223-247, 2000.
- [24] KANER, Cem; FALK, Jack; NGUYEN, Hung Quoc. *Testing computer software*. Dreamtech Press, 1999.
- [25] BEIZER, Boris. *Black-box testing: techniques for functional testing of software and systems*. John Wiley & Sons, Inc., 1995.

- [26] ARNOWITZ, Jonathan; ARENT, Michael; BERGER, Nevin. Effective prototyping for software makers. Elsevier, 2010.
- [27] SEFELIN, Reinhard; TSCHELIGI, Manfred; GILLER, Verena. Paper prototyping-what is it good for?: a comparison of paper-and computer-based low-fidelity prototyping. In: Human factors in computing systems (CHI'03). 2003. p. 778-779.
- [28] NIELSEN, Jakob. Usability inspection methods. In: Conference companion on Human factors in computing systems. ACM, 1994. p. 413-414.
- [29] KANER, Cem; BACH, James; PETTICHORD, Bret. Lessons learned in software testing. John Wiley & Sons, 2008.
- [30] NIELSEN, Jakob. Usability engineering. Elsevier, 1994.
- [31] WHITMAN, Michael E.; MATTORD, Herbert J. Principles of information security. Cengage Learning, 2011.
- [32] MOLYNEAUX, Ian. The art of application performance testing: Help for programmers and quality assurance. O'Reilly Media, Inc. 2009.
- [33] KITCHENHAM, Barbara; CHARTERS, Stuart. Guidelines for performing systematic literature reviews in software engineering. Keele University, UK, v. 9, 2007.
- [34] WOHLIN, Claes. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th international conference on evaluation and assessment in software engineering. ACM, 2014. p. 38.
- [35] GOLDMAN, Alfredo; KATAYAMA, Eduardo. Retrato da comunidade acadêmica de métodos ágeis no brasil. In: Workshop Brasileiro de Métodos Ágeis (WBMA 2011). 2011. p. 1-10.
- [36] FILHO, Maurício C. et al. Um Estudo de Caso sobre o Aumento de Qualidade de Software em Projetos de Sistemas de Informação que Utilizam Test Driven Development. In: Simpósio Brasileiro de Sistemas de Informação (SBSI). 2012.
- [37] GOUVEIA, Victor; JÚNIOR, Nelson; FURTADO, Ana; JUNIOR, Ivaldir; FURTADO, Felipe. Avaliando as métricas ágeis utilizadas pelas empresas certificadas no nível F do MPS.Br. In: Workshop Anual do MPS (WAMPS 2016). 2016. 85–93.

- [38] LUCENA, Percival; TIZZEI, Leonardo. Applying Software Craftmanship Practices to a Scrum Project: an Experience Report. In: Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software (WASHES 2016), 73-84.
- [39] FOWLER, Martin; BECK, Kent. Refactoring: improving the design of existing code. Addison-Wesley Professional, 1999.
- [40] DE CASTRO, Andreza MFV et al. Extension of selenium RC tool to perform automated testing with databases in web applications. In: Proceedings of the 8th International Workshop on Automation of Software Test. IEEE Press, 2013. p. 125-131.
- [41] VILLANES, Isabel Karina; COSTA, Erick Alexandre Bezerra; DIAS-NETO, Arilo Claudio. Automated mobile testing as a service (AM-TaaS). In: Services (SERVICES), 2015 IEEE World Congress on. IEEE. 2015. p. 79-86.
- [42] BRAZ, Alan; RUBIRA, Cecília Mary Fisher; VIEIRA, Marco. Development of Complex Software with Agile Method. In: Agile Conference (AGILE), 2015. IEEE. 2015. p. 97-101.
- [43] VAZ, Verônica T.; CONTE, Tayana; TRAVASSOS, Guilherme H. Empirical Assessments of a tool to support Web usability inspection. CLEI Electronic Journal, v. 16, n. 3, p. 6-6, 2013.
- [44] CABREJOS, Luis Jorge Enrique Rivero; KAWAKAMI, Guto; CONTE, Tayana Uchoa. Using a Controlled Experiment to Evaluate Usability Inspection Technologies for Improving the Quality of Mobile Web Applications Earlier in their Design. In: Software Engineering (SBES), 2014 Brazilian Symposium on. IEEE. 2014. p. 161-170.
- [45] NEVES, Cecília Eloy; CAMPOS, Fábio. Teste de Usabilidade do Módulo de Atendimento da Ferramenta Desktop de Gestão de Clínicas de Diagnóstico por Imagem, Uniclínica RIS. Blucher Engineering Proceedings, v. 3, n. 3, p. 1163-1180, 2016.
- [46] DE BARROS, Rafaela Queiroz et al. Análise de Usabilidade da Homepage de um Equipamento de Eletroencefalograma: Emotiv Epoc. Blucher Design Proceedings, v. 2, n. 1, p. 1374-1387, 2015.
- [47] COSTA, Ingrid et al. Melhor prevenir do que remediar: Avaliando usabilidade e UX de software antes de levá-lo para a sala de aula. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2016. p. 806.
- [48] FALCAO, Christianne; LEMOS, Ana Catarina; SOARES, Marcelo. Evaluation of natural user interface: A usability study based on the leap motion device. In: Conference on Applied

Human Factors and Ergonomics (AHFE 2015). Procedia Manufacturing, v. 3. 2015 p. 5490-5495.

- [49] GIL, Antonio Carlos. Métodos e técnicas de pesquisa social. Editora Atlas SA, 2008.
- [50] BABBIE, Earl. Métodos de pesquisas de survey. Ed. da UFMG, 1999.
- [51] SCHWABER, Ken; SUTHERLAND, Jeff. The scrum guide. Scrum Alliance, v. 21, 2011.
- [52] APPELO, J. Agile Record The Magazine for Agile Developers and Agile Testers. 2011.
- [53] GOMES, Fábio. TDD: fundamentos do desenvolvimento orientado a testes, 2013. Disponível em <<http://www.devmedia.com.br/tdd-fundamentos-do-desenvolvimento-orientado-a-testes/28151>> Acesso em: 29 jun. 2017
- [54] MENA-CHALCO, Jesús Pascual; JUNIOR, Cesar; MARCONDES, Roberto. ScriptLattes: an open-source knowledge extraction system from the Lattes platform. Journal of the Brazilian Computer Society, v. 15, n. 4, p. 31-39, 2009.

Apêndice A: Script de seleção de estudos semi-automatizada

Neste apêndice, é apresentada a transcrição do script para seleção semi-automatizada de publicações do relatório do ScriptLattes, escrito em Python 3.6.0. Este código-fonte e o relatório do ScriptLattes utilizado nesta pesquisa estão disponíveis em <https://gist.github.com/JoaoGFarias/b120c05c98a79cd3a6360f52ca3e9bdd>.

```
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
import csv
import re
import string
import yaml

def save_new_authors(authors, file_title='new_authors_list.txt'):
    authors.sort()
    with open(file_title, mode='w+') as file_handler:
        for item in authors:
            file_handler.write("{}\n".format(item))

def save_authors(authors, file_title='new_authors.yml'):
    with open(file_title, mode='w+') as file_handler:
        yaml.dump(authors, file_handler)

def save_papers(data, new_iteration_file='new_iteration.yml'):
    with open(new_iteration_file, mode='w+') as file_handler:
        yaml.dump(data, file_handler)

def clean_string(text):
    exclude = set(string.punctuation)
    return re.sub(' +', ' ', ''.join(ch for ch in text if ch not in exclude)).lower()
```

```

def should_paper_pass(title, paper_list):
    title = clean_string(title)

    if clean_string(title) in paper_list:
        return False

    response = input('Paper title: ' + title + '\nResponse (y/n):')
    return response == 'y'

def get_authors(file='all_authors.txt'):
    try:
        with open(file, newline='') as csvfile:
            reader = csv.reader(csvfile, delimiter=';', quotechar='|')
            return set([row[0] for row in reader])
    except FileNotFoundError:
        return set()

def loop_papers(driver, links):
    all_authors = get_authors('all_authors.txt')
    new_authors = []
    paper_list = []
    new_papers = {'papers':[]}
    for link in links:
        driver.get(link)
        works = driver.find_elements_by_css_selector('tr')
        for work in works:

            title = work.find_element_by_css_selector('td:nth-of-type(2) b').text

            #Escaping repeated works
            if clean_string(title) in paper_list:
                continue

            try:
                doi = work.find_element_by_css_selector('td:nth-of-type(2) a').get_attribute("href")
            except NoSuchElementException:
                continue

            authors = work.find_element_by_css_selector('td:nth-of-type(2) div').text

            if should_paper_pass(title, paper_list):

```

```

paper_list.append(clean_string(title))
work_authors = [x.strip() for x in authors.split(';')]
new_papers['papers'].append({
    'status':"",
    'title':clean_string(title),
    'doi': doi,
    'authors': work_authors})
new_authors.extend(work_authors)

authors_list = {'authors':[]}

#Excluding from already analyzed authors from the list of new authors
new_authors = set(new_authors).difference(all_authors)

for author in new_authors:
    authors_list['authors'].append({'name':author,'papers':[]})

save_new_authors(list(new_authors))

for paper in new_papers['papers']:
    for author in authors_list['authors']:
        if author['name'] in paper['authors']:
            author['papers'].append({'title':paper['title'],'doi':paper['doi']})

save_papers(new_papers)
save_authors(authors_list)
return driver

def start(links, path_to_phantomJs_executable):
    driver = webdriver.PhantomJS(executable_path=path_to_phantomJs_executable)
    driver = loop_papers(driver, links)
    driver.quit()

#Fill in the links for the ScriptLattes report pages where the papers are displayed
links = []

#Fill in the path for the PhantomJS executable on your machine
path_to_phantomJs_executable = "

start(links, path_to_phantomJs_executable)

```

Apêndice B: Lista de Estudos Primários

Neste apêndice, são apresentados os estudos primários selecionados para o mapeamento sistemático da literatura descrito no capítulo 3, e os resultados do processo de extração de dados.

A Tabela 2 lista os estudos primários selecionados pelo estudo do capítulo 3. Para cada estudo, foi determinado um identificador, que será usado no restante do apêndice, e seu o ano de publicação.

Tabela 2: Lista de estudos primários selecionados

ID	ANO	REFERÊNCIA
PS01	2013	Vaz, V.; Conte, T.; Travassos, G. Empirical Assessments of a tool to support Web usability inspection.
PS02	2013	Gomes, M.; Oliveira, H.; Conte, T. Evaluating a Usability Inspection Technique by means of Industry: Case Studies.
PS03	2015	Valentim, N.; Silva, W.; Conte, T. Avaliando a experiência do usuário e a usabilidade de um aplicativo web móvel: Um relato de experiência.
PS04	2012	Filho, M; Vasconcelo, J.; Santos, W. ; & Silva, I. Um Estudo de Caso sobre o Aumento de Qualidade de Software em Projetos de Sistemas de Informação que Utilizam Test Driven Development.
PS05	2014	Noronha, E.; Freitas, D. Prioritization of Artifacts for Unit Testing Using Genetic Algorithm Multi-objective Non Pareto.
PS06	2013	De Castro, A.; Macedo, G.; Collins, E.; Dias-Neto, A. Extension of Selenium RC tool to perform automated testing with databases in web applications.
PS07	2016	Camacho, C.; Marczak, S.; Cruzes, D. Agile Team Members Perceptions on Non-Functional Testing Influencing Factors from an Empirical Study.
PS08	2014	Xavier, B.; Lacerda, G.; Ribeiro, V.; Ribeiro, E.; Silveira, S.; Zabadal, J. Agile Data: Automating database refactorings. Journal of Engineering Research and Applications
PS09	2014	Rivero, L.; Kawakami, G.; Conte, T. Using a controlled experiment to evaluate usability inspection technologies for improving the quality of mobile web applications earlier in their design.
PS10	2014	Rivero, L.; Kalinowski, M.; Conte, T. Practical findings from applying innovative design usability evaluation technologies for mockups of web applications.
PS11	2012	Collins, E.; Vicente, V.; De Lucena, A. Strategies for Agile software testing automation: An industrial experience.
PS12	2012	Collins, E.; Macedo, G.; Maia, N.; Dias-Neto, A. An Industrial Experience on the Application of Distributed Testing in an Agile Software Development Environment.
PS13	2016	Neves, C.; Campos, F. Teste de Usabilidade do Módulo de Atendimento do UniClinika RIS: Ferramenta desktop para gestão de clínicas de diagnóstico por imagem
PS14	2012	Melo, C.; Santana, C.; Kon, F. Developers motivation in agile teams.
PS15	2012	Collins, E.; Lucena, V. Software Test Automation practices in agile development environment: An industry experience report.

ID	ANO	REFERÊNCIA
PS16	2015	Fontana, R.; Reinehr, S.; Malucelli, A. Agile Compass: A Tool for Identifying Maturity in Agile Software-Development Teams.
PS17	2012	Penedo, J.; Diniz, M.; Ferreira, S.; Silveira, D.; Capra, E. Análise de Usabilidade de um Sistema de EaD Baseada em Modelos Markovianos e em Taxonomia.
PS18	2012	Penedo, J.; Diniz, M.; Ferreira, S.; Silveira, D.; Capra, E. Evaluation of usability utilizing Markov models.
PS19	2015	Barros, R.; Monteiro, R.; Marcio, M. Análise de Usabilidade da Homepage de um equipamento de eletroencefalograma: Emotiv Epoc
PS20	2016	Nascimento, I.; Gaspar, W.; Conte, T.; Gadelha, B.; Elaine, H. Melhor prevenir do que remediar : Avaliando usabilidade e UX de software antes de levá-lo para a sala de aula.
PS21	2015	Santos, I.; Franco, W.; Sabóia, B.; Rossana, M. Definição e Aplicação de um Processo de Testes Ágeis : um Relato de Experiência.
PS22	2013	Ribeiro, S.; Albuquerque, A.; Filho, L.; Júnior, L. Uma avaliação da abordagem TDD em uma empresa desenvolvedora de software madura.
PS23	2012	Maia, N.; Macedo, G.; Collins, E.; Dias-neto, A. Aplicando Testes Ágeis com Equipes Distribuídas : Um Relato de Experiência.
PS24	2016	Gouveia, V.; Júnior, N.; Furtado, A.; Junior, I.; Furtado, F. Avaliando as métricas ágeis utilizadas pelas empresas certificadas no nível F do MPS.Br.
PS25	2014	Fontana, R.; Fontana, I.; Garbuio, P. ; Reinehr, S.; Malucelli, A. Processes versus people: How should agile software development maturity be defined?
PS26	2015	Albuquerque, D; Cafeo, B.; Garcia, A.; Barbosa, S.; Abrahão, S.; Ribeiro, A. Quantifying usability of domain-specific languages: An empirical study on software maintenance.
PS27	2015	Falcao, C.; Lemos, A.; Soares, M. Evaluation of Natural User Interface: A Usability Study Based on the Leap Motion Device.
PS28	2016	Souza, M.; Ribeiro, R.; Oliveira, P.; Neto, P. UseSkill: Uma abordagem para apoiar avaliações de usabilidade em sistemas web com base em mineração de dados.
PS29	2015	Vacari, I. Um estudo empírico sobre a adoção de métodos ágeis para desenvolvimento de software em organizações públicas.
PS30	2016	Freitas, E. SCOUT : A Multi-objective Method to Select Components in Designing Unit Testing.
PS31	2016	Santos, C.; Novais, T.; Albuquerque, C. Metrics focused on usability ISO 9126 based.

ID	ANO	REFERÊNCIA
PS32	2015	Valentim, N.; Conte, T.; Estacio, B.; Prikladnicki, R. How do software engineers apply an early usability inspection technique? A qualitative study.
PS33	2015	Cabrejos, L.; Vincenzi, A.; Maldonado, J.; Conte, T. Evaluating Software Engineers' Acceptance of a Technique and Tool for Web Usability Inspection
PS34	2016	Rodrigues, A.; Dias-Neto, A. Relevance and Impact of Critical Factors of Success in Software Test Automation lifecycle: A Survey.
PS35	2015	Braz, A.; Rubira, C.; Vieira, M. Development of Complex Software with Agile Method.
PS36	2016	Araújo, M.; Silva, A.; Darin, T.; Andrade, R.; Sánchez, J.; Lima, E. Design and Usability of a Braille-based Mobile Audiogame Environment.
PS37	2015	Furtado, A. Agile Practices in Maturity Model for Testing: an Experience Report
PS38	2016	Lucena, P.; Tizzei, L. Applying Software Craftmanship Practices to a Scrum Project: an Experience Report.
PS39	2015	Villanes, I.; Costa, E.; Dias-Neto, A. Automated Mobile Testing as a Service (AM-TaaS)
PS40	2015	Valentim, N.; Silva, W.; Conte, T. Avaliando a experiência do usuário e a usabilidade de um aplicativo web móvel: Um relato de experiência

Fonte: Elaborado pelo autor

A Tabela 3 mostra o mapeamento entre as técnicas de testes ágeis e os estudos primários relacionados.

Tabela 3: Mapeamento das técnicas nos estudos primários seleccionados

TÉCNICA	QUADRANTE	ESTUDOS PRIMÁRIOS
Cobertura de Testes	Q1	PS29
Métricas de Código	Q1	PS25
Revisão de Código	Q1	PS25; PS16; PS38
Complexidade Ciclomática	Q1	PS29
Complexidade Eferente	Q1	PS29
Testes de Integração	Q1	PS25; PS38
Programação em Pares	Q1	PS25; PS216; PS21
Análise Estática	Q1	PS25; PS38; PS35; PS24
TDD	Q1	PS37; PS25; PS16; PS38; PS04; PS22
Testes Unitários	Q1	PS25; PS16; PS38; PS21; PS05; PS11; PS15; PS30
TFA	Q2	PS06; PS11; PS23; PS35; PS21; PS15; PS23
BDD	Q2	PS37
Integração Contínua	Q2	PS11; PS23; PS15; PS08; PS16
Prototipagem	Q2	PS32; PS33
Inspeção de Usabilidade	Q2	PS01; PS10; PS17; PS18; PS19; PS20; PS09
Testes Automatizados de Usabilidade	Q3	PS28
Testes em Pares	Q3	PS21
Quantidade de Defeitos	Q3	PS24
Testes de Usabilidade	Q3	PS02; PS03; PS13; PS20; PS26; PS27; PS31; PS36; PS40
Testes de Performance	Q4	PS15; PS38
Testes de Segurança	Q4	PS15

Fonte: Elaborado pelo autor

Apêndice C – O Questionário

Este apêndice apresenta a transcrição do questionário aplicado nos profissionais de *software* do Porto Digital do Recife, descrito no capítulo 4. A pesquisa está disponível em <https://goo.gl/forms/zxZamlCWmDb7IB6o1>

Seção 1 - Apresentação

Testes Ágeis no Porto Digital

Meu nome é João Farias, sou aluno da graduação em Ciência da Computação da Universidade Federal de Pernambuco e estou desenvolvendo o meu trabalho de conclusão de curso (TCC) sob orientação do professor Alexandre Vasconcelos.

Essa pesquisa visa entender melhor a prática de testes em projetos ágeis no Porto Digital do Recife. Se você já trabalhou em algum momento neste contexto, você está convidado a responder essa pesquisa.

De acordo com testes, o preenchimento desse questionário deve durar entre 10 e 15 minutos. Se você tiver qualquer dúvida em relação à pesquisa, por favor, entre em contato comigo por meio do e-mail: jgfd@cin.ufpe.br

Termo de Consentimento Livre e Esclarecido (TCLE)

Sua participação não é obrigatória. A qualquer momento, você poderá desistir de participar e retirar seu consentimento. Sua recusa, desistência ou retirada de consentimento não acarretará prejuízo. Os dados obtidos por meio desta pesquisa serão confidenciais e não serão divulgados em nível individual, visando assegurar o sigilo de sua participação. A pesquisador responsável se compromete a tornar públicos nos meios acadêmicos e científicos os resultados obtidos de forma consolidada sem qualquer identificação de indivíduos ou instituições participantes. Caso você concorde em participar desta pesquisa, selecione o checkbox de “Concordo”.

Concordo

Seção 2

Contexto de Projeto

Qual sua experiência com projetos de software?

- Até 6 meses
- Até 2 anos
- Até 5 anos
- Até 10 anos
- Mais de 10 anos

Qual sua experiência com projetos de software que utilizaram metodologias ágeis?

- Até 6 meses
- Até 2 anos
- Até 5 anos
- Até 10 anos
- Mais de 10 anos

Quais papéis você desempenhou nos projetos ágeis?

- Testador
- Desenvolvedor
- Designer / Especialista de UX
- Gerente de Projeto
- Product Owner
- Scrum Master

Outro Qual o tamanho da(s) empresa(s) em que você trabalhou em projetos ágeis?

- Até 20 funcionários
- Até 50 funcionários
- Até 100 funcionários
- Mais de 100 funcionários

Qual o tipo de companhia de software você trabalhou?

- Baseada em Projetos
- Baseada em Produtos
- Fábricas de Software
- Serviços
- Consultoria
- Não sei informar

Qual o tamanho da(s) equipe(s) em que você trabalhou em projetos ágeis?

- Até 5 membros
- Até 10 membros
- Até 20 membros
- Mais de 20 membros

Havia uma equipe de testes dedicada?

- Sim
- Não, os membros do time de desenvolvimento eram responsáveis por testar.

Caso tenha respondido "Sim" à questão anterior, de que forma ocorria a integração entre a equipe de testes e de desenvolvimento?

- Todos os testadores trabalhavam in loco integrados com a equipe de desenvolvimento.
- Todos os testadores trabalhavam remotamente integrados com a equipe de desenvolvimento.
- Parte da equipe de testadores trabalhava in loco e outra parte trabalhava remotamente, porém

todos estavam integrados com a equipe de desenvolvimento.

Qual área do(s) projeto(s) utilizou-se de metodologias ágeis?

- Todo o time
- Apenas o time de desenvolvimento
- Apenas o time de testes

- Não foram utilizadas metodologias ágeis

Qual contexto dos projetos desenvolvidos com metodologias ágeis?

- Desenvolvimento Web
 Desenvolvimento Mobile
 Pesquisa Acadêmica
 Desenvolvimento de Hardware
 Suporte Técnico
 Manutenção de Software
 IoT
 Outro

Qual modelo de maturidade era seguido?

- CMM / CMMI
 ISO 25000
 ISO 9126-1
 MPS . BR
 MPT . BR
 Nenhum modelo de maturidade era seguido
 Outro

Seção 3 - Os Quadrantes de Testes Ágeis

Nas próximas seções serão indagadas questões relativas ao modelo de Quadrantes de Testes Ágeis desenvolvido por Lisa Crispin e Janet Gregory, onde as atividades de teste em projetos ágeis são divididas em quatro categorias, de acordo com os objetivos de cada teste. Para cada uma das quatro categorias, serão perguntados quais técnicas foram utilizadas nos projetos em que você trabalhou e qual sua percepção sobre o valor que estes testes trouxeram.

- Entendi, quero continuar

Seção 4 - Primeiro Quadrante

Quais técnicas foram utilizadas?

- Testes Unitários
- Testes de Componente
- Testes de Integração
- Análise Estática de Código
- Revisão de Código
- Programação em Pares
- TDD
- Este tipo de teste não foi realizado
- Outro

Quais papéis desenvolveram e executaram esse tipo de teste?

- QA / Testador
- Desenvolvedor
- Designer / UX
- Cliente / Product Owner

Com que frequência este tipo de teste foi aplicado?

- Nunca
- Eventualmente
- Frequentemente

Em que etapa do(s) projeto(s) esse tipo de teste foi aplicado?

- No início (até 25% do tempo total de duração do projeto)
- No meio (até 75% do tempo total de duração do projeto)
- No final (após 75% do tempo total de duração do projeto)

Qual foi a contribuição desse tipo de teste para o projeto?

- Essencial para o projeto

- Positiva
- Neutra
- Negativa
- Atrapalhou o projeto

Comentários sobre esse tipo de teste

Seção 5 - Segundo Quadrante

Quais técnicas foram utilizadas?

- Testes Funcionais Automáticos
- Prototipagem
- Simulação
- Behaviour Driven Development
- Acceptance Test Driven Development
- Este tipo de teste não foi realizado
- Outro

Quais papéis desenvolveram e executaram esse tipo de teste?

- QA / Testador
- Desenvolvedor
- Designer / UX
- Cliente / Product Owner

Com que frequência este tipo de teste foi aplicado?

- Nunca
- Eventualmente
- Frequentemente

Em que etapa do(s) projeto(s) esse tipo de teste foi aplicado?

- No início (até 25% do tempo total de duração do projeto)
- No meio (até 75% do tempo total de duração do projeto)
- No final (após 75% do tempo total de duração do projeto)

Qual foi a contribuição desse tipo de teste para o projeto?

- Essencial para o projeto
- Positiva
- Neutra
- Negativa
- Atrapalhou o projeto

Comentários sobre esse tipo de teste

Seção 6 - Terceiro Quadrante

Quais técnicas foram utilizadas?

- Testes Exploratórios
- Testes de Usabilidade
- Testes de Aceitação com o Usuário
- Testes Alpha/Beta
- Este tipo de teste não foi realizado
- Outro

Quais papéis desenvolveram e executaram esse tipo de teste?

- QA / Testador
- Desenvolvedor
- Designer / UX

Cliente / Product Owner

Com que frequência este tipo de teste foi aplicado?

Nunca

Eventualmente

Frequentemente

Em que etapa do(s) projeto(s) esse tipo de teste foi aplicado?

No início (até 25% do tempo total de duração do projeto)

No meio (até 75% do tempo total de duração do projeto)

No final (após 75% do tempo total de duração do projeto)

Qual foi a contribuição desse tipo de teste para o projeto?

Essencial para o projeto

Positiva

Neutra

Negativa

Atrapalhou o projeto

Comentários sobre esse tipo de teste

Sua Resposta

Seção 7 - Quarto Quadrante

Quais técnicas foram utilizadas?

Testes Performance

Testes de Segurança

Testes de Estresse

Testes de Carga

Testes de Escalabilidade

- Testes de Consistência
- Este tipo de teste não foi realizado
- Outro

Quais papéis desenvolveram e executaram esse tipo de teste?

- QA / Testador
- Desenvolvedor
- Designer / UX
- Cliente / Product Owner

Com que frequência este tipo de teste foi aplicado?

- Nunca
- Eventualmente
- Frequentemente

Em que etapa do(s) projeto(s) esse tipo de teste foi aplicado?

- No início (até 25% do tempo total de duração do projeto)
- No meio (até 75% do tempo total de duração do projeto)
- No final (após 75% do tempo total de duração do projeto)

Qual foi a contribuição desse tipo de teste para o projeto?

- Essencial para o projeto
- Positiva
- Neutra
- Negativa
- Atrapalhou o projeto

Comentários sobre esse tipo de teste

Sua Resposta

Seção 8 - Percepções sobre Testes Ágeis

Qualidade do Produto

Comparando com projetos que não utilizam metodologias ágeis nas atividades de teste, você considera que a qualidade dos produtos desenvolvidos em projetos que utilizam metodologias ágeis nas atividades de teste é:

1 ▼ to 5 ▼

1 Muito pior

5 Muito melhor

Engajamento do Time

Comparando com equipes que não utilizam metodologias ágeis nas atividades de teste, você considera que o engajamento dos membros das equipes que utilizam metodologias ágeis nas atividades de teste é:

1 ▼ to 5 ▼

1 Muito pior

5 Muito melhor

Quais os desafios percebidos com o uso de testes ágeis?

- Tempo de Transição
- Gerenciamento de Complexidade
- Habilidade Técnicas
- Habilidade de Comunicação

- Mudança de Cultura Organizacional
- Colaboração do Cliente
- Suporte Gerencial
- Redução de Custos
- Resistência a Mudança nas Equipes
- Outro