

**UNIVERSIDADE DE PASSO FUNDO**  
**INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA**

**CPFT**

**Uma proposta de processo adaptável para  
testes funcionais utilizando crowdsourcing**

**Mateus Henrique Dal Forno**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Computação Aplicada na Universidade de Passo Fundo.

**Orientador: Prof. Dr. Juliano Tonezer da Silva**  
**Coorientador: Prof. Me. Alexandre Lazaretti Zanatta**

Passo Fundo  
2016

**Substitua esta página pela  
Ficha Catalográfica  
(obrigatório para  
dissertações, inexistente em  
monografias)**

**Substitua esta página pelo  
Termo de Apresentação  
(obrigatório para  
dissertações, inexistente em  
monografias)**

*Dedico este trabalho aos meus pais, pelo apoio incondicional e exemplo de vida e dedicação.*

## AGRADECIMENTOS

Aos meus familiares agradeço pelo apoio incondicional, pelo estímulo nas horas difíceis e pela compreensão nos momentos de ausência, necessários para que a realização deste trabalho fosse possível.

Ao meu orientador, prof. Juliano Tonezer da Silva e ao meu co-orientador, prof. Alexandre Lazaretti Zanatta, agradeço pela compreensão, paciência, sabedoria, dedicação e colaboração na concretização deste trabalho. Suas críticas, considerações e sugestões foram de grande importância nessa difícil jornada.

A Universidade de Passo Fundo agradeço pelo apoio financeiro e material e incentivo à pesquisa.

As empresas de desenvolvimento de software de Passo Fundo agradeço por se prontificarem a ceder suas instalações, equipamentos, software e pessoal para a realização do estudo de caso, a parceria de vocês foi fundamental na avaliação do processo proposto neste trabalho.

Aos colegas de mestrado agradeço pelo apoio mútuo e pelas discussões, críticas e sugestões sobre nossas pesquisas.

A todos, que de uma maneira ou outra contribuíram na realização deste trabalho, meu muito obrigado.

*“Prefiro acreditar que, se forem persistentes, até pequenas gotas d’água, com o tempo, podem mudar uma pedra para sempre. E a pedra nunca voltará ao que era.”*

**(Veronica Roth)**

# CPFT

## RESUMO

A área de testes de software vêm tornando-se cada vez mais relevante, devido à crescente necessidade de ampliar a confiabilidade do software. Apesar da importância, há pouca utilização de testes nas empresas de pequeno porte, em que a demanda de trabalho para um testador é variável. Este trabalho apresenta a construção de um processo para a execução de testes funcionais utilizando crowdsourcing. O desenvolvimento do trabalho ocorreu em quatro etapas: Na primeira etapa realizou-se o estudo de abordagens de teste, bem como se investigou o uso do crowdsourcing para a realização de teste de software. A segunda etapa envolveu a imersão em uma empresa de desenvolvimento de software, onde realizou-se a implantação do processo de teste de Sommerville, buscando identificar quais atividades do processo poderiam ser realizadas utilizando-se o crowdsourcing. A terceira etapa envolveu a modelagem do processo de teste. Buscou-se estabelecer uma estrutura adaptável, que permita a integração ao processo de desenvolvimento de software que a empresa já utiliza. Além disso, o processo também é adaptável quanto aos artefatos, podendo a empresa adequá-los aos artefatos que já utiliza, ou substituí-los por ferramentas ou softwares. O processo é composto por três fases: Pré-crowd, na qual ocorre o planejamento e elaboração dos casos de teste; Crowd, que é responsável pela disponibilização e execução das tarefas de teste e; Pós-crowd, que envolve a avaliação do resultado reportado a cada tarefa de teste e avaliação/registo das falhas identificadas. A última etapa envolveu a avaliação do processo, onde optou-se pela abordagem de estudo de caso. O estudo de caso realizado é do tipo descritivo e integrado, composto por duas unidades de análise, cada uma relativa à uma empresa de desenvolvimento de Software da cidade de Passo Fundo. Estabeleceu-se quinze proposições, que foram avaliadas com os dados coletados a partir da execução do processo em cada unidade de análise. Destas, nove foram confirmadas, cinco permaneceram inconclusivas e uma foi refutada. A análise dos resultados obtida com o estudo de caso evidencia a viabilidade do processo de teste proposto, e aponta questões que permanecem inconclusivas.

Palavras-Chave: Teste de Software, Crowdsourcing, Modelagem de processo.

# CPFT

## ABSTRACT

The software testing area is becoming more popular because of the software's reliability. There are a few software testing areas in small business, due to a poor tester labor. This paper presents the construction of a process for performing functional tests through crowdsourcing. The study was developed in four stages: at the first stage, the test approaches study was carried out, as well as the investigation of the use of crowdsourcing for the performance of software test. The second step involved the immersion into a software house, where the Sommerville test process was implemented, identifying the process activities which could be performed via crowdsourcing. The third step involved the process development proposed. The aim was to establish an adaptable structure, one which enables the integration of the process to the software development process used by the organization. In addition, the process is also adaptable regarding the artifacts and the company can adapt them to the equipment which is being used, or replace them by tools or software. The three stages process: Pre-crowd, when the planning and preparation of test cases occur; Crowd, which is responsible for the provision and performance of test tasks and; Post-crowd, which involves the evaluation of the result reported in each test task and evaluation and registration of identified faults. The last stage involved the process evaluation, when it was decided to use the case study approach. The case study is descriptive and integrated, with two analysis units, each one regarding one software house from Passo Fundo city. Fifteen propositions were established, and they were evaluated with the data collected from the performance of the process in each analysis unit. Nine propositions were confirmed, five remain inconclusive and one was refuted. The results show that it evinces the feasibility of the test process proposed, and identifies issues which remain inconclusive.

Keywords: Software Testing, Crowdsourcing, Process Modeling.

## LISTA DE FIGURAS

Figura 1.	Etapas propostas para a execução do trabalho. Fonte: elaborada pelo autor. ....	18
Figura 2.	Relação entre os elementos fundamentais no crowdsourcing. Fonte primária. ....	28
Figura 3.	Taxonomia de qualidade em sistemas crowdsourcing. Fonte: adaptado de [24, p. 77]. ....	34
Figura 4.	Processo proposto por Keimel et al. Fonte: adaptado de [25, p. 246].	36
Figura 5.	Processo de teste de interfaces por meio do Mturk. Fonte: adaptado de [30, p. 334]. ....	38
Figura 6.	Modelo de processo de teste. Fonte: Sommerville [4, p.147] . . . . .	42
Figura 7.	Diagrama BPMN do processo. Fonte: elaborada pelo autor. ....	44
Figura 8.	Organização dos elementos na documentação do processo. Fonte: elaborada pelo autor. ....	47
Figura 9.	Exemplo de atividade. Fonte: elaborada pelo autor. ....	48
Figura 10.	Exemplo de ator. Fonte: elaborada pelo autor. ....	49
Figura 11.	Exemplo de artefato. Fonte: elaborada pelo autor. ....	50
Figura 12.	Roteiro para implantação do processo. Fonte: elaborada pelo autor.	51
Figura 13.	Fase 1 - Pré-crowd. Fonte: elaborada pelo autor. ....	54
Figura 14.	Fase 2 - Crowd. Fonte: elaborada pelo autor. ....	57
Figura 15.	Fase 3 - Pós-crowd. Fonte: elaborada pelo autor. ....	60

## LISTA DE TABELAS

Tabela 1.	Situação das proposições após o estudo de caso. ....	84
-----------	--	----

## LISTA DE SIGLAS

ABNT – Associação Brasileira de Normas Técnicas  
BPM – Business Process Management  
BPMN – Business Process Model and Notation  
CPFT – Crowdsourcing Process for Functional Tests  
DPN – Diagrama de Processos de Negócio  
ERP – Enterprise Resource Planning  
EPF COMPOSER – Eclipse Process Framework Composer  
GQM – Goal Question Metric  
HP – Hewlett-Packard  
HTML – HyperText Markup Language  
IDE – Integrated Development Environment  
IEEE – Institute of Electrical and Electronics Engineers  
ISO – International Organization for Standardization  
MCTI – Ministério da Ciência, Tecnologia e Inovação  
MTM – Microsoft Test Manager  
OMG – Object Management Group  
P2P – Peer-to-Peer  
PDCA – Plan–Do–Check–Act  
SGBD – Sistema de Gerenciamento de Banco de Dados  
TIC – Tecnologias da Informação e Comunicação  
UML – Unified Modeling Language  
UPF – Universidade de Passo Fundo  
URL – Uniform Resource Locator  
VPN – Virtual Private Network  
XML – eXtensible Markup Language

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	CONTEXTO E MOTIVAÇÃO	15
1.2	OBJETIVOS	17
1.2.1	<b>Objetivo geral</b>	17
1.2.2	<b>Objetivos específicos</b>	17
1.3	METODOLOGIA DE TRABALHO	17
1.4	ORGANIZAÇÃO DO DOCUMENTO	20
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>21</b>
2.1	PROCESSO	21
2.1.1	<b>Business Process Management Notation</b>	22
2.2	TESTE DE SOFTWARE	23
2.2.1	<b>Teste Funcional de Software</b>	26
2.3	CROWDSOURCING	27
2.3.1	<b>O uso de Crowdsourcing para teste de software</b>	30
2.4	CONSIDERAÇÕES SOBRE O CAPÍTULO	32
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>34</b>
3.1	QUALITY CONTROL IN CROWDSOURCING SYSTEMS	34
3.2	QUALITYCROWD - A FRAMEWORK FOR CROWD-BASED QUALITY EVALUATION	36
3.3	CROWDSOURCING GUI TESTS	37
3.4	CONSIDERAÇÕES SOBRE O CAPÍTULO	40
<b>4</b>	<b>CROWDSOURCING PROCESS FOR FUNCTIONAL TESTS - CPFT</b>	<b>42</b>
4.1	CONSTRUÇÃO DO PROCESSO	42
4.2	ATORES DO PROCESSO	52
4.2.1	<b>Analista de testes</b>	52
4.2.2	<b>Analista da plataforma crowdsourcing</b>	52
4.2.3	<b>Multidão de testadores</b>	53
4.3	FASES DO PROCESSO E ATIVIDADES	53
4.4	FASE 1 - PRÉ-CROWD	54

4.4.1	<b>Atividade 1 - Elaborar plano de teste crowdsourcing</b>	54
4.4.2	<b>Atividade 2 - Definir os cenários de teste</b>	55
4.4.3	<b>Atividade 3 - Elaborar os casos de teste</b>	56
4.5	FASE 2 - CROWD	56
4.5.1	<b>Atividade 4 - Configurar ambiente de teste</b>	57
4.5.2	<b>Atividade 5 - Especificar Tarefa de Teste</b>	58
4.5.3	<b>Atividade 6 - Executar Tarefa de Teste</b>	58
4.5.4	<b>Atividade 7 - Submeter resultado da tarefa de teste</b>	59
4.6	FASE 3 - PÓS-CROWD	59
4.6.1	<b>Atividade 8 - Avaliar execução da tarefa de teste</b>	60
4.6.2	<b>Atividade 9 - Avaliar Falhas Reportadas</b>	61
4.7	ARTEFATOS DO PROCESSO	61
4.7.1	<b>Documento de requisitos</b>	61
4.7.2	<b>Plano de teste crowdsourcing</b>	61
4.7.3	<b>Cenários de teste</b>	62
4.7.4	<b>Caso de teste</b>	62
4.7.5	<b>Tarefa de teste</b>	62
4.7.6	<b>Resultado da tarefa de teste</b>	62
4.7.7	<b>Registro de falha</b>	63
4.8	CONSIDERAÇÕES SOBRE O CAPÍTULO	63
<b>5</b>	<b>ESTUDO DE CASO</b>	<b>64</b>
5.1	APRESENTAÇÃO DO ESTUDO PILOTO	64
5.2	APRESENTAÇÃO DA CONDUÇÃO DO ESTUDO DE CASO EM CADA UNIDADE DE ANÁLISE	70
5.2.1	<b>Fontes de evidência utilizadas</b>	70
5.2.2	<b>Unidade de análise I</b>	71
5.2.3	<b>Unidade de análise II</b>	75
5.3	ANÁLISE E DISCUSSÃO DAS PROPOSIÇÕES	79
5.4	CONSIDERAÇÕES SOBRE O CAPÍTULO	84
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>86</b>
6.1	DIFICULDADES ENCONTRADAS	88
6.2	TRABALHOS FUTUROS	88
	<b>REFERÊNCIAS</b>	<b>90</b>

<b>APÊNDICE A</b> – Projeto de pesquisa .....	<b>93</b>
<b>APÊNDICE B</b> – Protocolo para o estudo de caso .....	<b>95</b>
<b>APÊNDICE C</b> – Levantamento inicial .....	<b>98</b>
<b>APÊNDICE D</b> – Roteiro de avaliação .....	<b>99</b>

# 1. INTRODUÇÃO

É notória a importância do software na vida das pessoas. Sua presença é implícita em tarefas simples do cotidiano como por exemplo, o uso de um caixa eletrônico para a retirada de dinheiro. Também, em condições críticas que exigem alta disponibilidade e confiabilidade, como por exemplo em softwares de controle de tráfego aéreo. Apesar da importância e da ubiquidade do software, existem empresas de desenvolvimento de software que não executam testes nos softwares que comercializam e, desta forma, não há comprovação de que estes funcionam conforme o esperado. É neste contexto que o presente trabalho é introduzido, demonstrando a importância de um processo de teste como estratégia para a verificação e validação de um software.

Este Capítulo traz na Seção 1.1 o contexto e a motivação, demonstrando e justificando o problema de pesquisa. A Seção 1.2 apresenta o detalhamento dos objetivos deste trabalho. A Seção 1.3 apresenta a metodologia de trabalho aplicada. Por fim, a Seção 1.4 apresenta e caracteriza os demais Capítulos deste trabalho.

## 1.1 CONTEXTO E MOTIVAÇÃO

As Tecnologias da Informação e Comunicação (TICs) estão presentes nas atividades humanas, tornando-se essenciais para a execução de tarefas em diversas áreas e exigindo uma maior qualidade nos softwares desenvolvidos e utilizados. Cada vez mais é necessário que a qualidade dos softwares seja ampliada, devido ao crescimento de seu uso para o controle da execução de tarefas críticas [1]. Além da qualidade, o crescimento do uso das tecnologias nas atividades cotidianas amplia a expectativa dos usuários, que buscam a automação de processos, agilidade, segurança e confiabilidade nas soluções que utilizam.

Levando em consideração o uso de técnicas de teste descritas pela literatura [2, 3, 4] em empresas de desenvolvimento de software, observa-se que há uma baixa utilização das mesmas, bem como em alguns casos o seu uso ocorre de forma inadequada, apesar de sua importância como parte do processo de desenvolvimento de software. Dados do Ministério da Ciência, Tecnologia e Inovação (MCTI), presentes no relatório “Indicadores da Qualidade e Produtividade”<sup>1</sup> indicam que o percentual de empresas brasileiras que adotam teste de sistema nos softwares desenvolvidos é de apenas 45%.

Neste contexto, o trabalho de Rizzi [5] avaliou o uso das abordagens de teste pelas empresas de desenvolvimento de software em Passo Fundo, verificando se a realidade regional se assemelhava aos dados observados em nível nacional. Para isso, foi realizado um estudo em empresas da região, com o objetivo de investigar o conhecimento e a aplica-

---

<sup>1</sup>Disponível em <[http://www.mct.gov.br/index.php/content/view/47820/Tecnologia\\_de\\_Software.html](http://www.mct.gov.br/index.php/content/view/47820/Tecnologia_de_Software.html)>.

ção das abordagens de teste de software. Em relação ao nível de teste, das 17 empresas participantes da pesquisa, apenas sete (40%) afirmaram que realizam testes de sistema nos softwares que desenvolvem.

Outro dado relevante obtido com a pesquisa é que 10 empresas (59%) afirmaram utilizar testes funcionais, que é uma estratégia de teste a nível de sistema. Porém, destas empresas, apenas quatro (ou 24% do total) informaram a forma como o teste é realizado. Em relação aos critérios, três empresas declararam utilizar a técnica de análise do valor limite, bem como duas empresas afirmaram utilizar a técnica de particionamento em classes de equivalência, teste funcional sistemático e grafo causa-efeito. As demais não souberam informar a técnica que utilizam para a execução dos testes. Com base na contradição demonstrada por esses dados, pode-se inferir que a execução destes testes na maioria destas empresas ocorre de maneira informal.

Em contraponto à crescente necessidade de melhoria da qualidade [1], o trabalho de Rizzi [5] demonstra que há pouca utilização de testes de software por empresas da região de Passo Fundo. Bem como, destas empresas que realizam testes, especialmente o teste funcional, apenas uma pequena parcela segue as recomendações e o formato apresentado pela academia. Desta forma, percebe-se que a realidade local não difere da realidade nacional em relação à aplicação de testes pelas empresas de desenvolvimento de software.

Uma das alternativas para ampliar o uso de teste de software pelas empresas é a adoção de estratégias que utilizam a inteligência coletiva e plataformas baseadas no conceito de tecnologia na nuvem [6]. Um estudo realizado em conjunto pela empresa de consultoria Capgemini, pela Sogeti e pela HP, disponível no relatório “World Quality Report” [7], indica uma tendência de aumento da adoção do uso de plataformas de teste disponíveis na nuvem, sugerindo que em 2015 tal abordagem de teste poderá representar cerca de um terço (32%) das atividades de teste realizadas no planeta. Tal dado demonstra a existência de um campo fértil para o estudo do uso das iniciativas baseadas na nuvem humana, bem como do crowdsourcing, para a realização de testes.

A motivação deste trabalho está na possibilidade de integrar o crowdsourcing a um processo de teste, sendo este utilizado como estratégia para a execução de testes funcionais. Acredita-se que esta abordagem seja possível, baseado em outros trabalhos que aplicaram o crowdsourcing para a realização de testes que validam características não funcionais de um software, tais como interface, usabilidade e desempenho [8].

## 1.2 OBJETIVOS

### 1.2.1 Objetivo geral

Desenvolver um processo de teste funcional baseado em crowdsourcing.

### 1.2.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

1. Realizar estudo sobre teste de software funcional;
2. Realizar estudo sobre o uso de crowdsourcing no processo de desenvolvimento de software;
3. Realizar estudo sobre o uso de crowdsourcing no processo de teste de software;
4. Propor um processo de teste funcional baseado em crowdsourcing para pequenas software houses;
5. Avaliar o processo proposto, por meio da execução do processo em um estudo de caso.

## 1.3 METODOLOGIA DE TRABALHO

Este trabalho se enquadra, quanto à metodologia, como sendo uma pesquisa tecnológica, que é um tipo de pesquisa científica aplicada, obtendo como produto final, neste caso, um processo. Quanto à forma de abordagem, esta pesquisa é qualitativa, ou seja, possui caráter descritivo e enfoque indutivo [9].

Em relação aos procedimentos, esta pesquisa é classificada como experimental, onde são provocadas alterações no ambiente-alvo da pesquisa, e as intervenções realizadas são observadas, em busca de identificar se estão produzindo os resultados esperados [10].

O desenvolvimento deste trabalho foi organizado em quatro etapas: A etapa de revisão buscou o estudo das abordagens de teste existentes, bem como a investigação do uso de crowdsourcing para o teste de software. A segunda etapa envolveu a imersão do autor em uma empresa de desenvolvimento de software, a fim de observar a sistemática do ambiente de trabalho desta empresa e implantar o processo de teste proposto por Sommerville [4]. Na terceira etapa, elaborou-se um processo de teste funcional utilizando

crowdsourcing. Por fim, na quarta etapa, houve a avaliação do trabalho realizado por meio de um estudo de caso.

As etapas propostas para a execução deste trabalho são descritas em detalhes a seguir. A Figura 1 ilustra as quatro etapas que compõem os passos metodológicos.

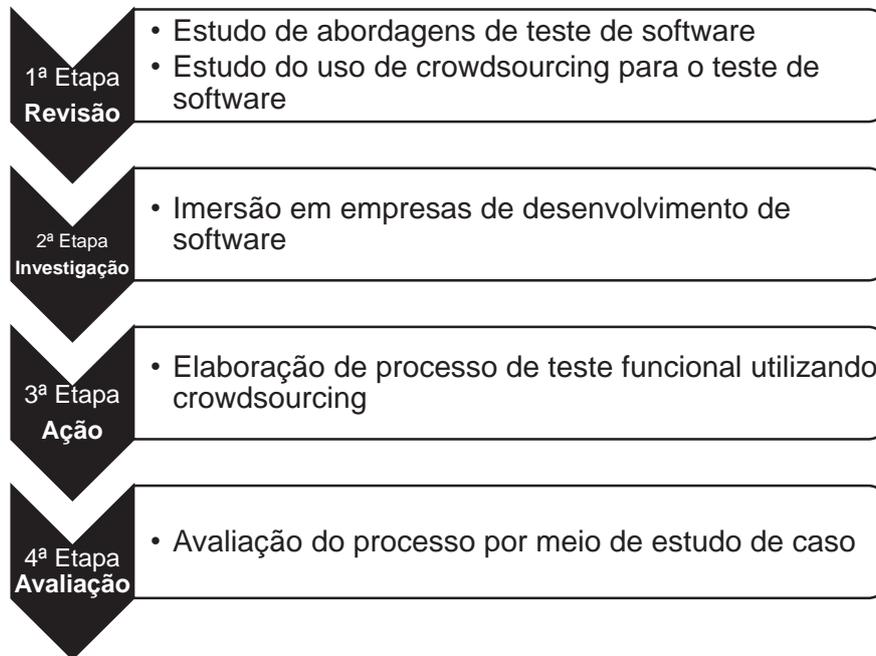


Figura 1. Etapas propostas para a execução do trabalho. Fonte: elaborada pelo autor.

### 1ª Etapa - Revisão

A etapa de Revisão é composta por duas atividades. Uma das atividades envolveu a revisão do conhecimento em teste de software, compreendendo seus modelos, técnicas, níveis e abordagens, de forma a revisar e estabilizar conceitos. A realização desta atividade envolveu a leitura de trabalhos realizados na área, bem como de referências bibliográficas.

A segunda atividade envolveu o estudo de crowdsourcing, um tema emergente, especialmente na área da computação. Para a execução desta atividade, realizou-se a revisão bibliográfica e o levantamento de trabalhos relacionados à temática de inteligência coletiva e crowdsourcing. Em relação ao crowdsourcing, buscou-se o aprofundamento com foco em questões inerentes ao seu uso para o desenvolvimento e teste de software. Adicionalmente, realizou-se o levantamento de plataformas crowdsourcing para o desenvolvimento e teste de software, com foco na identificação de iniciativas de crowdsourcing existentes, bem como em sua sistemática de funcionamento.

## 2ª Etapa - Investigação

Na segunda etapa, denominada como “investigação”, houve a imersão em uma empresa de desenvolvimento de software. Nesta imersão, observou-se a sistemática do ambiente de trabalho da empresa, buscando identificar os problemas existentes na estratégia de teste adotada pela empresa. Posteriormente, implantou-se o processo de teste proposto por Sommerville [4], o qual serviu como base para a elaboração do processo de teste proposto neste trabalho.

## 3ª Etapa - Ação

A terceira etapa envolveu o desenvolvimento do processo de teste, utilizando o crowdsourcing como estratégia para a execução dos casos de teste funcionais .

Para a representação do processo foram elaborados diagramas utilizando a notação de modelagem de processos de negócio (BPMN). Como forma de documentar as atividades, atores e artefatos do processo, utilizou-se o software EPF Composer <sup>2</sup>, que organiza a documentação do processo, realizando links entre as atividades e seus respectivos atores e artefatos relacionados, por fim gerando um website para consulta da documentação do processo.

## 4ª Etapa - Avaliação

A última etapa proposta envolveu a avaliação do processo de teste proposto. Para a avaliação da proposta, optou-se pelo uso do estudo de caso como método, composto de um caso único e duas unidades integradas de análise.

Optou-se pelo uso da abordagem de estudo de caso devido à impossibilidade de total controle do ambiente e dos eventos comportamentais do processo, especialmente em relação à execução do mesmo, que ocorre em ambiente externo. Esta característica do processo impossibilita a adoção de técnicas de avaliação que necessitam ser utilizadas em um ambiente controlado como, por exemplo, a engenharia de software experimental.

Para a execução do estudo de caso, utilizou-se como referência a abordagem descrita por Yin [11]. O estudo de caso proposto é do tipo descritivo, em que o foco está em descrever determinada intervenção e o contexto em que a mesma ocorreu, ilustrando a avaliação realizada. Quanto às unidades de análise, o estudo de caso é do tipo integrado, em que há um contexto único, composto por duas unidades integradas de análise, sendo cada uma correspondente à uma empresa de desenvolvimento de software.

---

<sup>2</sup>Direitos reservados à Eclipse Foundation, disponível em <<https://eclipse.org/epf/>>.

É importante ressaltar que o foco do estudo de caso é qualitativo, ou seja, não representa uma “amostragem quantitativa”, mas apresenta uma generalização analítica, ao mesmo tempo que expande teorias [11] (neste caso, demonstrando que o processo pode ser executado em uma pequena empresa de desenvolvimento de software).

#### 1.4 ORGANIZAÇÃO DO DOCUMENTO

A organização deste trabalho é descrita a seguir. O Capítulo 2 apresenta a fundamentação teórica, trazendo conceitos e definições relacionadas ao tema deste trabalho. A Seção 2.1 apresenta conceitos relacionados a processo, bem como do BPMN, técnica de modelagem de processos de negócio utilizada neste trabalho. Na Seção 2.2 são apresentados conceitos relacionados ao teste de software, bem como ao teste funcional, técnica de teste executada pelo processo proposto neste trabalho. Por fim, a Seção 2.3 apresenta o conceito de crowdsourcing, bem como sua aplicação para teste de software.

O Capítulo 3 apresenta e descreve trabalhos relacionados ao tema deste trabalho, descrevendo um trabalho que caracteriza atributos de qualidade em uma plataforma crowdsourcing, bem como trabalhos que propõem processos de teste de software utilizando crowdsourcing para a avaliação de streaming de vídeo e de interface.

No Capítulo 4 é apresentado o Crowdsourcing Process for Functional Tests - CPFT. Na Seção 4.1 é descrita a construção do processo, a partir do modelo de processo de teste de Sommerville [4]. A Seção 4.2 apresenta os atores responsáveis por executar as atividades do processo. A Seção 4.3 apresenta e descreve as três fases do processo: pré-crowd, crowd e pós-crowd, bem como suas respectivas atividades. Por fim, a Seção 4.7 apresenta os artefatos desenvolvidos para apoiar a execução das atividades do processo.

O relato da condução do estudo de caso é descrito no Capítulo 5. A Seção 5.1 apresenta a execução do estudo de caso piloto. Na Seção 5.2 cada unidade de análise do estudo de caso é descrita, juntamente com a condução do estudo de caso. As proposições são analisadas e discutidas na Seção 5.3, em que são levantadas evidências que confirmam ou refutam as hipóteses inicialmente propostas.

Por fim, o Capítulo 6 apresenta as considerações finais, apresentando e discutindo as contribuições realizadas neste trabalho. Na Seção 6.1 são descritas as dificuldades encontradas durante o estudo. Encerrando o Capítulo, a Seção 6.2 apresenta os trabalhos futuros.

## 2. FUNDAMENTAÇÃO TEÓRICA

Este Capítulo apresenta a fundamentação teórica. Na Seção 2.1 são apresentados conceitos relacionados a processo, bem como a BPMN, técnica de modelagem de processos de negócio utilizada neste trabalho. A Seção 2.2 apresenta conceitos relacionados a teste de software, bem como a teste funcional, técnica de teste executada pelo processo proposto neste trabalho. Por fim, a Seção 2.3 apresenta o conceito de crowdsourcing, bem como sua aplicação para teste de software.

### 2.1 PROCESSO

A palavra “processo” é originada do termo em latim “*procedere*”, que é conceituado como uma maneira de agir ou um conjunto de ações tomadas para atingir determinado objetivo [12]. Neste trabalho será adotada a definição da International Organization for Standardization (ISO), que define processo como um “conjunto de atividades inter-relacionadas ou interativas, que transformam entradas em saídas” [13, p. 3]. A ISO afirma que utilizar processos é uma maneira poderosa de organizar e gerenciar as atividades, criando valor para o cliente e os stakeholders [13].

Carli [12] considera que processos sempre existiram, utilizando como exemplo o homem pré-histórico, que executava determinada sequência de ações para encurralar sua caça, ou então para produzir fogo, que incluía critérios para a escolha dos gravetos e técnicas adequadas para gerar fricção e calor. Carli afirma que a forma de pensamento “Entrada-processamento-saída” originou-se no fordismo, (produção em linha de montagem idealizada por Henry Ford), em que as atividades industriais aconteciam de maneira padronizada, seguindo uma ordem definida e conhecida pelos que as executavam, o que permitia que pudessem ser replicáveis e controláveis.

O objetivo do uso dos processos é contribuir para a sistematização da estrutura de uma organização, servindo como instrumento de ligação de tudo o que se faz, facilitando a comunicação e a cooperação e servindo de elo entre as estratégias e as atividades diárias, seu conhecimento, seus recursos e suas ferramentas [14].

O uso dos processos de negócio busca o máximo de produtividade e o menor índice de erros e desperdícios, focando na produção de um resultado conhecido. Os processos são fundamentais para que as organizações forneçam produtos e serviços com preços competitivos, reduzindo custos e despesas e maximizando lucros [12]. Modelar um processo visa entender e repensar a organização da empresa, buscando assegurar que todos os colaboradores possuam a mesma visão do negócio, padronizando conceitos e visões, melhorando a qualidade e a produtividade dos produtos e serviços [14].

### 2.1.1 Business Process Management Notation

A Business Process Management Notation é uma técnica para representação de processos de negócio, com foco na modelagem, na análise e na orquestração, sendo atualmente uma das técnicas mais difundidas e largamente aceita para esta finalidade [14]. É importante ressaltar a diferença entre BPMN e BPM. BPM (Business Process Management) é uma disciplina gerencial para a gestão de processos, enquanto a BPMN é a notação para a representação dos processos de negócio.

A BPMN permite a modelagem de processos dos mais variados tipos e natureza, como por exemplo: processos administrativos, financeiros, operacionais, de garantia da qualidade ou de desenvolvimento de software, podendo ser utilizada para comunicar informações a uma grande variedade de públicos [14].

Além disso, a BPMN é um padrão para modelagem de processos, voltado para a definição e documentação de processos de negócio, com notação de elementos definida. Possui um único modelo de diagrama, chamado de Diagrama de Processos de Negócio (DPN), que possibilita o desenho de diversos tipos de modelagem de processos. Este diagrama é composto por quatro elementos: atividades, eventos, decisões (gateways) e conectores [14].

O elemento **atividade** representa um trabalho que será executado no processo de negócio, podendo ser uma tarefa ou um subprocesso [14]. A atividade é considerada como uma tarefa quando é atômica, ou seja, não pode ser representada em um nível inferior de detalhamento. Um subprocesso é uma atividade que é composta por duas ou mais tarefas ou atividades.

Um **evento** é algo que acontece durante a execução do processo, afetando seu fluxo, podendo ser de início, intermediário ou de fim [14]. Eventos de início indicam onde um processo começa, afetando o fluxo do processo e sendo disparados em determinado contexto ou situação. Eventos intermediários ocorrem entre eventos de início e de fim, afetando o fluxo do processo, mas não iniciando-o ou terminando-o. Eventos de fim indicam onde o processo irá encerrar, geralmente possuindo um resultado.

Elementos de decisão, ou **gateways**, são responsáveis por controlar a sequência do fluxo dentro do processo, separando e unindo fluxos, baseados em dados ou em eventos [14].

Os **Fluxos** interligam os elementos do processo, podendo ser de sequência, de mensagens ou de associação [14]. Fluxos de sequência indicam a ordem que as atividades são executadas no processo e também interligam atividades a eventos e gateways. Fluxos de mensagem são utilizados para mostrar transições de mensagem entre duas entidades. Por fim, associações são utilizadas para vincular dados, informações e artefatos aos elementos do processo.

A BPMN possui também outros elementos, que podem ser utilizados conforme as necessidades específicas do processo a ser modelado. Um exemplo são as raias [14], que são utilizadas em situações em que é necessário representar papéis (entidades ou participantes) que estão separados fisicamente, especificando as atividades associadas a cada papel. Outro exemplo são os objetos de dados, que representam artefatos que são requeridos ou produzidos pelas atividades, sendo conectados a estas por meio das associações.

## 2.2 TESTE DE SOFTWARE

O teste de software é uma atividade que permeia as demais atividades do processo de desenvolvimento de software, desde a etapa de levantamento de requisitos até a evolução do software. O processo de teste possui vida própria dentro do processo de desenvolvimento de software e pode ocorrer em paralelo com as demais atividades presentes [3]. Sommerville [4] afirma que o teste possui dois objetivos distintos:

- Demonstrar que o software atende aos requisitos;
- Identificar situações onde o software comporta-se de maneira inesperada, incorreta ou de forma diferente da definida na especificação de requisitos.

Neste trabalho será adotada a definição de teste descrita por Myers [15]. Myers afirma que teste “é o processo de executar um software com o objetivo de encontrar erros” [15, p. 5, tradução nossa]. Neste contexto, o teste é parte do processo de verificação e validação e possui o papel de identificar e demonstrar a presença de erros em um software, e não a sua ausência [4].

A atividade de teste deve ser planejada de forma que o software possa fornecer o nível de confiabilidade adequado no contexto em que se encontra inserido. Com o uso cada vez maior de softwares para o controle de atividades críticas, como a operação de usinas nucleares e o controle aeroespacial, é necessário que seja demandado um grande esforço de testes nesses softwares, pois pequenos defeitos podem gerar consequências que colocam em risco a vida das pessoas [1].

Os testes podem ser classificados em níveis, técnicas e tipos. Considerando o nível, os testes podem ser classificados em testes de unidade, testes de integração e testes de sistema. Em relação ao teste de unidade (ou teste unitário), Sommerville [4] o conceitua como sendo o nível de teste que tem como objetivo testar os métodos ou funções individuais. Pressman [2] afirma que o teste unitário possui enfoque na lógica de processamento interna e nas estruturas de dados presentes nos limites de um componente, buscando garantir que as informações fluam corretamente pela unidade que está sendo testada.

O teste de integração busca avaliar se os componentes do software estão corretamente integrados, bem como se os componentes se comunicam e interagem da forma esperada, atingindo os objetivos previstos [4, 2]. Bartié [16] afirma que os testes de integração são uma continuação natural dos testes unitários, pois as unidades construídas ou modificadas devem manter a compatibilidade com os demais componentes existentes e até então integrados.

De acordo com Pressman [2], teste de sistema é o nome dado a um conjunto de testes que possuem como finalidade exercitar todo o software. Este conjunto de testes verifica se os elementos do software estão corretamente integrados e executando as devidas funções aos quais foram projetados. Sommerville [4] afirma que os testes baseados em casos de uso são eficazes para a realização de testes de sistema, devido a forçarem as interações entre os componentes ou objetos do software.

Em relação à classificação de acordo com a técnica, os testes podem ser classificados como caixa preta ou caixa branca. Quanto à abordagem caixa preta, Koscianski e Soares [17] afirmam que o nome “caixa preta” originou-se da engenharia, e que nesta abordagem, ao se analisar o comportamento de um objeto, sua estrutura interna é ignorada. Por meio do teste, são fornecidos os dados de entrada e são observadas as saídas produzidas, verificando se as mesmas correspondem ao esperado [3]. Myers [15] considera que na abordagem caixa preta o testador deve buscar circunstâncias em que o software não se comporta conforme as especificações.

De acordo com Pressman [2], testes caixa branca preocupam-se com a estrutura lógica do software e fundamentam-se em exames rigorosos dos detalhes procedimentais. Nesta abordagem, os testes devem ser planejados de forma a exercitar as estruturas internas, em busca do maior número possível de erros [16]. Koscianski e Soares [17] afirmam que o fato do testador ter acesso aos códigos permite que sejam projetados testes mais precisos.

Uma terceira classificação organiza os testes de acordo com a sua categoria. Por meio desta classificação é possível direcionar os esforços de teste em uma determinada perspectiva. De acordo com Bartié [16], existem 12 categorias de teste:

- **Teste Funcional (ou de Funcionalidade):** Tem por objetivo simular cenários de negócio e avaliar se o comportamento do software é condizente com o definido na especificação de requisitos;
- **Teste de Usabilidade:** Tem como finalidade simular a utilização do software sob a perspectiva do usuário final, preocupando-se com o nível de facilidade de uso e se o mesmo atende aos requisitos de usabilidade definidos;
- **Teste de Carga (Stress):** Visa simular condições de uso atípicas do software, oscilando as transações até o limite máximo de transações sucessivas e avaliando como o software se comporta em situações onde ocorrem variações de processamento;

- **Teste de Volume:** Diferentemente do teste de carga, tem por objetivo determinar o limite de carga da infraestrutura, por meio do aumento contínuo dos parâmetros de execução até o limite suportável pelo software em operação;
- **Teste de Configuração:** Tem como finalidade a execução do software em ambientes diferentes, de forma a avaliar a compatibilidade do mesmo nas configurações de software e hardware especificadas;
- **Teste de Compatibilidade:** Tem por objetivo verificar a interação do software com versões anteriores, sejam elas tanto de software, quanto de hardware, ou de ambas;
- **Teste de Segurança:** Tem como foco detectar falhas que comprometem o sigilo dos dados do software, bem como as que permitam alteração de informações e interrupções de processamento;
- **Teste de Performance:** Visa avaliar se o software cumpre satisfatoriamente com os requisitos de desempenho definidos em situações em que o volume de transações está no limite especificado;
- **Teste de Instalação:** Tem por objetivo testar os procedimentos de instalação do software, bem como verificar se a instalação ocorre de acordo com o especificado nos requisitos, sendo recomendado que o usuário final realize este tipo de teste;
- **Teste de Confiabilidade e Disponibilidade:** Tem por finalidade realizar o monitoramento do software por um período, verificando o nível de confiabilidade do mesmo, bem como o tempo de disponibilidade e o tempo de indisponibilidade (necessário para que o mesmo seja restabelecido);
- **Teste de Recuperação:** Busca avaliar o comportamento do software na ocorrência de uma condição anormal, especialmente em softwares de missão crítica, em que é necessário um alto índice de disponibilidade;
- **Teste de Contingência:** Tem por objetivo validar os procedimentos de contingência aplicados pela equipe de suporte em situações especificadas.

Neste trabalho, optou-se pelo teste funcional devido à sua importância como técnica de teste. Bartié [16, p. 113] afirma que o teste funcional é a “categoria mais prioritária dentre as demais, pois representa a aderência do software em relação às regras de negócio”, ressaltando a importância da utilização desta categoria de teste como forma de garantir que os requisitos funcionais do software foram efetivamente implementados da forma correta, bem como assegurar que o comportamento do software é o esperado.

Adicionalmente, Mao et al [8] realizaram uma revisão sistemática do uso de crowdsourcing na engenharia de software. Em relação ao teste de software, foram identificadas

iniciativas que utilizam o crowdsourcing como estratégia para execução de testes que validam aspectos não funcionais de um software: usabilidade, desempenho e interface. A ausência de estudos do uso de crowdsourcing para as demais categorias de teste, demonstrada pelos autores, indica campos de estudo ainda inexplorados, dentre eles o teste funcional.

### 2.2.1 Teste Funcional de Software

De acordo com Myers [15], teste funcional é uma técnica que localiza discrepâncias entre um software e uma especificação do comportamento esperado. Fabbri, Vincenzi e Maldonado [18] afirmam que trata-se de uma técnica de teste sob o ponto de vista do usuário, na qual o software é considerado como uma caixa preta; e, cujos casos de teste são baseados na informação de campos de entrada definidos e análise das saídas obtidas com saídas esperadas. Bartié [16] acrescenta que os testes funcionais devem simular os cenários de negócio e garantir que todos os requisitos funcionais sejam implementados, bem como a inexistência de diferenças entre o comportamento do software e os requisitos funcionais.

A princípio, a técnica de teste funcional permite a execução de testes para todas as entradas possíveis, o que é denominado como teste exaustivo. Porém, como geralmente o domínio de dados de entrada é muito grande, torna-se inviável a realização de testes exaustivos em softwares. Em função disso, definiu-se critérios de teste que, se executados de maneira sistemática, permitem que haja uma cobertura de testes adequada, desde que exista uma especificação de requisitos [18].

O particionamento em classes de equivalência é um critério de teste que divide o domínio de entrada de dados em classes de dados, para permitir posteriormente a elaboração dos casos de teste [2, 16]. Neste critério, utiliza-se um subconjunto de todas as entradas possíveis, selecionando o subconjunto com a maior probabilidade de localizar a maioria dos erros, reduzindo o número de casos de teste ao necessário para se alcançar uma cobertura adequada e cobrindo um vasto conjunto de outros possíveis casos de teste [15]. A primeira etapa nesta abordagem é a identificação das classes de equivalência, a partir de cada uma das entradas e a organização das mesmas em grupos. Posteriormente, são definidas classes válidas (que representam entradas válidas para o software) e classes inválidas (que representam os estados inválidos possíveis) [15, 16]. A partir dessas classes são extraídos os casos de teste, até que todas as classes de equivalência estejam cobertas por casos de testes [15].

A análise do valor limite é um método complementar ao particionamento em classes de equivalência [2, 16]. Na técnica de análise do valor limite são consideradas as situações acima e abaixo das bordas das classes de equivalência, onde cada extremidade

é objeto de teste [15]. Myers [15] afirma que casos de testes que exploram valores limite têm melhores resultados e que, se praticada corretamente, é uma das técnicas mais úteis.

O teste funcional sistemático é um critério de teste que utiliza conjuntamente os critérios definidos pelo particionamento de classes de equivalência e a análise do valor limite [18]. Para isso, o teste funcional sistemático requer que pelo menos sejam elaborados dois casos de teste para cada partição, de forma a evitar que defeitos coincidentes mascarem falhas, bem como também é necessário que sejam avaliados o limite de cada partição e o seu limite subsequente [18].

O grafo causa-efeito é uma técnica que auxilia a definição de um conjunto de dados para possibilitar a exploração de ambiguidades e incompletudes nas especificações de teste [15, 18]. Trata-se de uma linguagem formal, que traduz a especificação de casos de teste da linguagem natural para a representação em formato de grafo, com notação própria. Posteriormente, a partir da representação obtida pelo grafo, são derivados os casos de teste. A vantagem desta abordagem é que a mesma exercita combinações de dados que possivelmente não seriam consideradas [18].

Error guessing é a técnica em que aplica-se uma abordagem de intuição, a partir da experiência, para a identificação de alguns prováveis erros e a partir disso derivar os casos de testes a fim de detectá-los [15, 18]. Myers [15] afirma que nesta técnica o analista de testes utiliza inconscientemente um dos critérios existentes para a derivação de casos de testes elaborando uma lista de possíveis erros ou situações que gerem erros, e a partir desta lista são gerados os casos de teste.

## 2.3 CROWDSOURCING

O termo “crowdsourcing” surgiu no ano de 2006, sendo cunhado e utilizado pela primeira vez pelo jornalista norte-americano Jeff Howe, no artigo intitulado “*The rise of crowdsourcing*”, publicado na revista Wired [19]. Howe conceitua crowdsourcing como sendo o ato de uma organização tomar uma tarefa, que era até então realizada internamente, e terceirizar a mesma a uma grande rede de pessoas, por meio de um convite aberto. O crowdsourcing desenvolveu-se rapidamente na última década, sendo impulsionado pelo advento da Web 2.0 [20, 21].

O surgimento das atividades que caracterizam o crowdsourcing é anterior à definição do próprio termo. Howe descreve em seu livro [22] que o crowdsourcing manifestou-se a partir das ações de milhares de pessoas que realizam a prática de *hobbies*, em companhia de pessoas com os mesmos gostos. A internet atua como o meio facilitador, que possibilita que as pessoas se comuniquem e se unam, buscando a realização de seus interesses. Além disso, o aprimoramento e o surgimento de novas tecnologias e as mudanças dos ecossistemas de trabalho vêm aprimorando esta forma de trabalho [23].

O modelo de produção que o crowdsourcing utiliza é dependente da inteligência coletiva, ou seja, do conhecimento e da experiência dos indivíduos conectados à internet, para solucionar diversos problemas. É uma maneira de realizar tarefas que podem ser resolvidas por seres humanos, porém são difíceis de serem solucionadas por computadores [24, 25]. O crowdsourcing pode propiciar às organizações o aumento da produtividade, a redução de custos e a melhoria dos processos de apoio [24]. Trata-se de um modelo de negócio que envolve terceirização colaborativa, podendo esta, ser remunerada ou não remunerada [20]. A realização das tarefas acontece de forma colaborativa entre os vários indivíduos ou pode ser realizada por apenas um deles [26].

Para compreender esta abordagem é importante definir e delimitar três elementos fundamentais no crowdsourcing, que são os clientes, as plataformas e a multidão [23]. A Figura 2 ilustra e resume a relação existente entre estes três elementos.



Figura 2. Relação entre os elementos fundamentais no crowdsourcing. Fonte primária.

No contexto do crowdsourcing, os clientes são indivíduos que possuem tarefas a serem realizadas, que podem envolver as mais diversas áreas, como por exemplo a elaboração de projetos de design gráfico, de desenvolvimento e testes de software, de criação de marketing/conteúdo criativo, vendas e suporte, dentre outros [27]. Em relação às vantagens para os clientes que utilizam esta abordagem, pode-se citar a facilidade da realização de tarefas “sob demanda”, ou seja, o indivíduo que realiza as tarefas para a organização não possui vínculo empregatício com a mesma, e a organização se beneficia de uma força de trabalho que é mobilizada de acordo com sua demanda. A organização realiza o pagamento pela execução da tarefa, e somente após sua conclusão. Desta forma, ocorre uma redução de custos, uma vez que não é necessário pagar pelo tempo demandado para que a tarefa seja cumprida, como ocorre nas organizações, em que os funcionários são pagos pela sua jornada de trabalho [28].

O segundo elemento existente são as plataformas crowdsourcing. As plataformas permitem e apoiam a mobilização da multidão, gerenciando e controlando a realização das atividades. As plataformas são mantidas por empresas, que são responsáveis pelo suporte

das plataformas. Existem também plataformas que se responsabilizam por aspectos técnicos da realização das tarefas crowdsourcing, incluindo-se a criação das atividades a serem desenvolvidas, a seleção dos indivíduos para sua realização, a coleta e envio dos resultados ao cliente e o pagamento pela realização das tarefas aos indivíduos (caso seja uma tarefa que envolva remuneração) [23, 27].

As plataformas crowdsourcing apresentam um mesmo conjunto de características, independentemente das especificidades em relação ao tipo de atividades que dão suporte [26]. Tais características envolvem:

- Uso de painéis administrativos nas plataformas, sendo utilizados pelos administradores da plataforma e pelos organizadores das atividades para o gerenciamento das tarefas;
- Ferramentas de colaboração e comunicação, que possibilitam a realização de discussões, com a participação dos indivíduos e os responsáveis pelas plataformas;
- Ranqueamento de participantes e ferramentas de recomendação;
- Ferramentas que possibilitam o desenvolvimento da tarefa solicitada;
- Gestão de pagamento por meio de plataformas de pagamento;
- Repositório de arquivos, disponibilizando informações que envolvem a especificação da tarefa, bem como critérios de qualidade e de aceitação da solução a ser desenvolvida;

O terceiro elemento do crowdsourcing é a multidão, que é composta por indivíduos que encontram-se dispersos geograficamente. O crowdsourcing utiliza a capacidade criativa desses indivíduos para a realização das tarefas solicitadas pelos clientes, que são disponibilizadas por meio das diversas plataformas existentes. Cada indivíduo realiza a tarefa de forma individual, tendo a iniciativa de selecionar as tarefas que deseja realizar, de acordo com seus gostos e interesses pessoais. O retorno obtido por meio da realização de determinada tarefa pode ocorrer pelo pagamento de determinado valor definido, e também em forma de reputação nas plataformas crowdsourcing, demonstrando o comprometimento do indivíduo na realização das tarefas e evidenciando o nível de qualidade de sua produção [20, 23, 29, 27].

Ainda existem algumas barreiras na utilização do crowdsourcing. Kaganer et al. [6] relatam que muitas organizações têm receio em delegar a realização de uma tarefa a pessoas externas ao seu ambiente, onde as negociações se realizam por meios virtuais. Devido a isso, as organizações procuram utilizar estratégias crowdsourcing apenas para a realização de projetos que possuam baixo orçamento e sem cronograma rígido para sua realização.

Além disso, as tarefas devem incluir todas as informações necessárias ao indivíduo que irá realizá-la, sob pena dos resultados obtidos com a execução da tarefa serem de baixa qualidade. Adicionalmente, a necessidade de conhecimentos e experiências específicos para a realização de algumas tarefas também é apontada pelos autores [24, 23].

Há ainda alguns riscos no uso do crowdsourcing. Um deles é o fracasso do projeto [6], que pode ocorrer devido ao possível desrespeito dos prazos ou desistência dos indivíduos em realizar determinada tarefa. Outro risco existente é o dos resultados obtidos para as tarefas serem superficiais e de baixa qualidade [21], devido à interpretações incorretas da atividade a ser realizada, bem como o desinteresse do indivíduo em se esforçar para a realização da mesma. Também é relevante nesta abordagem o risco de vazamento da propriedade intelectual da organização [6], uma vez que, para a realização de algumas tarefas, é necessária a divulgação de informações estratégicas para que seja possível a sua resolução.

### **2.3.1 O uso de Crowdsourcing para teste de software**

De acordo com Capgemini, Sogeti e HP [7, p. 32, tradução nossa] “Construir e manter um ambiente de teste pode ser difícil e caro”. Em alguns casos, tais ambientes ficam por muito tempo sem serem utilizados. Ambientes de testes devem simular com o máximo de realismo possível o ambiente real de uso do software em produção, sob pena da execução dos testes gerar resultados inconsistentes e distorcidos [7]. Em relação aos recursos humanos envolvidos, a atividade de teste possui uma demanda variável, ou seja, é oneroso para as organizações manterem grandes equipes de teste em virtude da sua variabilidade de demanda [30].

É neste contexto que surgiram as plataformas dedicadas a realização de testes. Por meio das plataformas de teste disponíveis, é possível a realização de testes com o uso de ferramentas e utilizando mão de obra sob demanda, de acordo com as necessidades específicas de cada projeto [7, 30]. Chen e Luo [31, p. 1, tradução nossa] ponderam que “um dos desafios do uso de crowdsourcing para testes é encontrar um número de trabalhadores qualificados e a um baixo custo”. Os autores consideram necessária também a realização de inspeções em resultados de testes obtidos, em virtude das habilidades irregulares dos testadores e também das possíveis fraudes na realização das tarefas crowdsourcing.

Em estudo realizado em conjunto pela empresa de consultoria Capgemini, pela empresa Sogeti, que pesquisa e desenvolve tecnologias para teste de software, e pela empresa multinacional Hewlett-Packard (HP), disponível no relatório “World Quality Report” [7], em sua quinta edição, indica uma queda no uso de plataformas de teste disponíveis na nuvem, de 28% em 2012 para 24% em 2013.

O relatório apresenta como justificativa para esta queda o fato das empresas estarem em uma fase de avaliação dos resultados obtidos com o uso dos testes na nuvem e

identificação dos ajustes necessários nos processos e nos ambientes de teste. Outra questão relevante e que está em discussão é relacionada à segurança e divulgação de dados da empresa para a realização dos testes e de seus projetos, que estarão disponíveis e visíveis para empresas concorrentes. Apesar destas questões, o estudo aponta que as empresas estão amadurecendo seus controles de qualidade e realizando modificações que permitirão no futuro uma maior adoção de ferramentas disponíveis na nuvem para a realização de testes. Uma das alternativas citadas no relatório envolve o desenvolvimento de softwares em que os ambientes de desenvolvimento e teste são abertos, porém com a posterior utilização em um ambiente local, ou dentro de uma *Virtual Private Network* (VPN)<sup>1</sup>.

Em entrevista conduzida pela Capgemini, pela Sogeti e pela HP [7], realizada com 1500 executivos de diversas empresas, os mesmos sugerem e indicam uma tendência de que os testes realizados por meio de softwares disponíveis na nuvem (dentre elas o crowdsourcing) representarão em 2015 um terço (32%) de todas as atividades de teste realizadas pelas empresas.

Em relação ao uso de crowdsourcing para a realização de teste de software, Tsai, Wu e Huhns [26] relatam que a Microsoft utilizou esta estratégia para a realização de testes, investindo US\$ 100.000 para localizar falhas de segurança durante o desenvolvimento do Windows 8.

No cenário nacional, a plataforma CrowdTest permite a realização de testes por meio de uma plataforma crowdsourcing. Em seu portal, é disponibilizada uma apresentação em que são descritas algumas características do uso de crowdsourcing para o teste de software<sup>2</sup>, as quais são descritas a seguir:

- Com o uso de crowdsourcing, várias pessoas são mobilizadas para a execução de testes e detecção de falhas. Este modelo de teste pode ser utilizado em situações em que há restrições de prazos para a realização dos testes e também em casos em que há recursos financeiros limitados para a realização dos testes;
- Com o convite aberto, é possível a mobilização rápida de uma equipe com pessoas de vários perfis distintos para a realização de testes. Cada indivíduo realiza os testes no horário que preferir, e para a organização não há a despesa com horas-extra e nem com o tempo demandado para a realização dos testes, uma vez que o pagamento é realizado por produtividade;
- Da mesma forma que ocorre em demais plataformas crowdsourcing, os indivíduos testadores recebem pela produtividade alcançada, envolvendo a identificação dos defeitos encontrados e, em alguns casos, pode envolver premiações por rendimentos alcançados;

---

<sup>1</sup> Rede virtual privada construída sobre a internet, criando canais criptografados para transferência de informações entre computadores e redes remotas.

<sup>2</sup> Disponível em <<http://crowdtest.me/utilizar-crowdsourcing-testar-software-palestra>>.

- Com o uso do modelo de crowdsourcing para testes é possível a realização de testes utilizando-se os ambientes de teste disponibilizados pelos testadores, incluindo-se navegadores, sistemas operacionais e dispositivos, em diversas versões, sem a necessidade de investimentos da empresa nestes ambientes de teste;
- Neste modelo de realização de testes é necessário que os defeitos encontrados sejam validados, com objetivo de certificar-se da qualidade dos resultados obtidos com os testes realizados pela multidão;
- É necessário que sejam definidos padrões de documentos que possibilitem a padronização da documentação dos resultados obtidos a partir dos testes;
- Da mesma forma que em outras áreas, a aplicação de crowdsourcing para testes impossibilita a garantia de sigilo do projeto em teste, pois pessoas desconhecidas têm acesso ao software. Desta forma, o crowdtest não é aplicável em qualquer cenário de testes;
- O uso de crowdsourcing para a realização de testes em softwares especializados é limitado, uma vez que em muitos casos o testador não possui conhecimento do negócio envolvido com o mesmo.

Uma das possibilidades do uso de testes por meio de crowdsourcing envolve a realização de testes de interface. Dolstra, Vliegendhart e Pouwelse [30] consideram que a interface gráfica de um software é a parte mais visível e a que o usuário possui maior contato. Devido a isto é importante a automatização da realização de testes de interface, o que pode garantir que mudanças realizadas no software não irão interromper a funcionalidade de determinado elemento da interface.

Um dos problemas relacionados ao teste de interfaces é em relação à dificuldade de sua realização. De acordo com Dolstra, Vliegendhart e Pouwelse [30, p. 332], “As interfaces gráficas são difíceis de serem testadas: testes automatizados são difíceis de serem criados e mantidos, enquanto os testes manuais são caros, demorados e difíceis de serem integrados em um processo de teste contínuo”. A elaboração de testes automatizados é complexa em virtude da fragilidade de sua especificação, em que pequenas alterações na interface geram falhas na execução dos testes, o que faz com que seja necessária a reescrita dos mesmos. Outra questão relevante neste processo é a de que testes de interface necessariamente devem ser realizados por seres humanos, pois é uma tarefa complexa e relativa para um computador definir se uma interface visual está corretamente elaborada ou não [30].

## 2.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Este capítulo apresentou a fundamentação teórica deste trabalho, apresentando conceitos e definições em relação a processo, teste de software e crowdsourcing.

Em relação ao tema “processo”, a primeira seção do capítulo apresentou seu conceito, bem como caracterizou o uso de processos de negócios nas organizações. Adicionalmente, apresentou-se a notação BPMN, bem como caracterizou-se seus elementos. A definição de processo utilizada neste trabalho descreve este como sendo um “conjunto de atividades inter-relacionadas ou interativas, que transformam entradas em saídas” [13, p. 3].

A seção sobre a temática “teste de software” apresentou o conceito de teste, bem como os níveis, técnicas e tipos de teste existentes. A definição de teste adotada neste trabalho é a de Myers, que define teste como o “processo de executar um software com o objetivo de encontrar erros” [15, p. 5, tradução nossa]. Adicionalmente, descreveu-se a técnica de teste funcional de software, bem como apresentou-se os critérios de teste, que proporcionam uma maior cobertura dos testes funcionais.

Por fim, na terceira seção, apresentou-se o conceito de “crowdsourcing”, bem como identificou-se e caracterizou-se os três elementos existentes nesta abordagem: o cliente, a plataforma e a multidão. Identificou-se algumas barreiras existentes no uso de crowdsourcing, como por exemplo a necessidade de definição de critérios para a seleção do indivíduo que irá realizar uma determinada tarefa, levando em consideração diferentes níveis de conhecimento e de habilidade, bem como a complexidade da tarefa a ser realizada. Por fim é apresentado o uso de crowdsourcing para o teste de software, fazendo a ligação entre estes dois elementos, bem como, apresentando algumas iniciativas já realizadas neste sentido.

### 3. TRABALHOS RELACIONADOS

Este capítulo apresenta trabalhos relacionados com a temática e o propósito deste trabalho. O trabalho apresentado na seção 3.1 descreve o estudo e o desenvolvimento de um modelo que caracteriza as dimensões de qualidade em sistemas crowdsourcing. As seções 3.2 e 3.3 apresentam trabalhos que propõem processos de teste de software, utilizando o crowdsourcing como estratégia para a execução dos testes. No primeiro trabalho, é descrita a realização de testes de *streaming* de vídeo e no segundo trabalho é relatado o uso do crowdsourcing para a realização de testes de interface gráfica.

#### 3.1 QUALITY CONTROL IN CROWDSOURCING SYSTEMS

Com o objetivo de desenvolver um modelo de controle de qualidade para o desenvolvimento de plataformas crowdsourcing, Allahbakhsh et al. [24] desenvolveram um modelo que caracteriza os atributos de controle de qualidade, organizando-os em duas dimensões distintas: perfil do trabalhador e projeto da tarefa. Como forma de identificar os atributos envolvidos em cada uma destas duas dimensões, os autores utilizaram a definição de qualidade de Crosby. A Figura 3 ilustra a taxonomia elaborada no trabalho.



Figura 3. Taxonomia de qualidade em sistemas crowdsourcing. Fonte: adaptado de [24, p. 77].

Uma das dimensões consideradas pelos autores é o perfil do trabalhador. Nesta dimensão, são considerados os atributos referentes à **reputação** e à **experiência** dos indivíduos que realizam as tarefas e as submetem nas plataformas crowdsourcing. Há uma

relação direta entre os atributos, pois um indivíduo com boa reputação na plataforma terá uma boa experiência no desenvolvimento das tarefas, e vice-versa. Apesar disso, pode-se considerar a reputação como uma métrica útil e que pode ser considerada para a avaliação do profissional, independente da atividade em questão. Já a experiência é considerada uma métrica específica de determinado tipo de atividade, pois a experiência na realização de tarefas de desenvolvimento não garante que o indivíduo terá experiência para realizar atividades de teste, por exemplo.

A segunda dimensão apresentada envolve questões relacionadas ao projeto da tarefa. Os autores relacionam a tarefa com quatro atributos. A **definição** da tarefa é o atributo que descreve o que deve ser realizado, bem como as restrições envolvidas. Neste atributo podem estar definidos também critérios de seleção para os indivíduos que poderão realizar a tarefa, podendo envolver, por exemplo, uma porcentagem mínima de trabalhos anteriores aceitos, ou restrições à participação de indivíduos de um determinado país. Adicionalmente, os autores descrevem que o nível de qualidade e a clareza das informações presentes na definição da tarefa interferem na qualidade de seu resultado.

A **interface de usuário** é o atributo que envolve a interface da plataforma crowdsourcing, pela qual os indivíduos terão acesso às tarefas e por onde submeterão o seu desenvolvimento. Neste contexto, é importante o uso de interfaces que sejam de fácil utilização, de forma a atrair mais indivíduos para a plataforma. Além disso, é importante que a plataforma possua formas de validar as submissões dos indivíduos, de forma a identificar tarefas incompletas e indivíduos que submetem qualquer resposta, fraudando a realização da tarefa apenas para receber a remuneração.

Em relação à dimensão de **granularidade**, é importante que as tarefas crowdsourcing sejam simples e independentes. Tarefas demasiadamente complexas levam um tempo maior para serem resolvidas e, em função disso, menos pessoas se interessam em realizá-las.

O quarto atributo definido para a dimensão de projeto da tarefa envolve a **política de remuneração**. Os autores consideram importante que as tarefas possuam remunerações que incentivem de forma adequada os indivíduos, o que fará com que surjam contribuições de melhor qualidade. Neste contexto, as remunerações podem ser intrínsecas ou extrínsecas. Remunerações intrínsecas incentivam o indivíduo a realizar uma tarefa em função de uma satisfação pessoal e as remunerações extrínsecas envolvem a recompensa monetária. Uma boa política de remuneração envolve ambas, porém uma remuneração monetária por si só não garante a qualidade do resultado da tarefa.

Ao final da realização do trabalho os autores consideraram que, apesar das várias abordagens de controle de qualidade existentes, é necessária a realização de pesquisas relacionadas à definição, medição e gerenciamento de qualidade em sistemas crowdsourcing.

As dimensões para caracterizar os atributos de controle de qualidade, descritas pelos autores, foram consideradas neste trabalho durante a condução do estudo de caso. A plataforma crowdsourcing selecionada para a execução dos testes foi avaliada quanto aos atributos definidos pelos autores, onde buscou-se identificar se os mesmos eram contemplados pela plataforma.

### 3.2 QUALITYCROWD - A FRAMEWORK FOR CROWD-BASED QUALITY EVALUATION

A avaliação da qualidade de transmissão de vídeo via streaming é subjetiva, pois não existem métricas de qualidade objetivas definidas e normatizadas. Além disso, é um processo de testes demorado e caro.

Neste contexto, o trabalho de Keimel et al. [25], que envolve a criação de um framework para a avaliação da qualidade de vídeo é descrito no artigo “*QualityCrowd - A Framework for Crowd-based Quality Evaluation*”. A Figura 4 ilustra o fluxo de trabalho para a realização dos testes, proposto pelos autores.

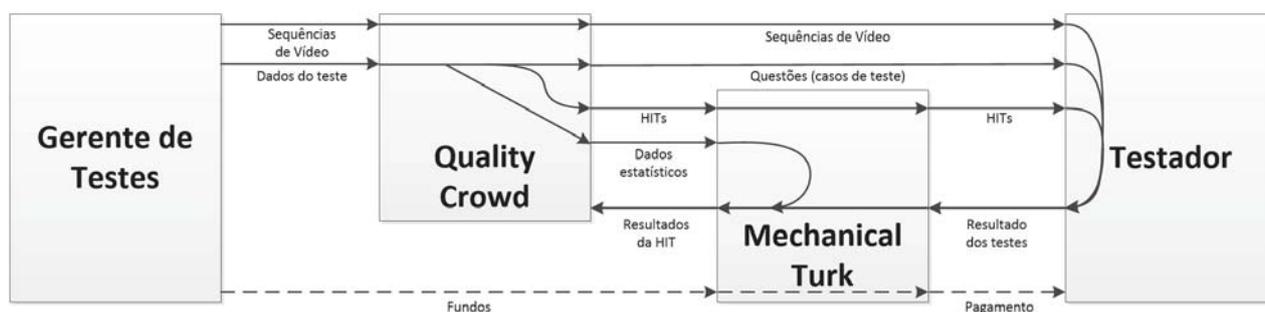


Figura 4. Processo proposto por Keimel et al. Fonte: adaptado de [25, p. 246].

Foi desenvolvido um framework para a realização dos testes, denominado de QualityCrowd. Sua arquitetura é composta por duas partes. Uma parte front-end, que o testador acessará para a realização dos testes, e uma parte back-end, que os gerentes de testes criam novos testes, realizam o upload de novos vídeos e gerenciam os testes já existentes. Este framework é uma software web, acessada por meio de navegadores. O framework utiliza a plataforma Amazon Mechanical Turk (Mturk) para divulgar as tarefas de teste a serem realizadas e também para realizar o pagamento ao testador.

O processo ocorre da seguinte forma: o gerente de testes inicialmente seleciona a sequência de vídeos que será testada e faz o seu upload no QualityCrowd. Juntamente com os vídeos, o gerente de testes define o tipo de teste e cria as perguntas. Após a conclusão da configuração do teste, o gerente de testes pode iniciar o teste de vídeo. Em seguida, a tarefa de teste é automaticamente criada na plataforma Mturk.

O testador, então, seleciona uma atividade de teste na plataforma Mturk e é redirecionado para o framework QualityCrowd, por onde são disponibilizados os vídeos a serem

testados e as perguntas previamente definidas. O testador realiza os testes e responde as perguntas solicitadas. Após a conclusão das atividades pelo testador, o QualityCrowd armazena os resultados obtidos em uma base de dados e envia uma estimativa de qualidade das respostas à plataforma Mturk. Por meio desta estimativa será definido se o testador receberá ou não o pagamento pela realização do teste. Esta etapa é importante, pois por meio deste procedimento é possível evitar fraudes na realização dos testes.

Para a avaliação desta proposta, os autores do trabalho utilizaram um conjunto de 28 vídeos, os quais foram submetidos anteriormente a testes realizados por meio do uso de um ambiente tradicional. Os resultados destes testes estavam disponíveis em um dos trabalhos citados pelos autores. Desta forma, pode-se realizar um comparativo dos resultados obtidos por ambas as abordagens. Os testes foram realizados por 19 testadores e optou-se pela realização dos mesmos em uma rede local em que o QualityCrowd está, com o objetivo de minimizar problemas de conexão à internet. Os resultados obtidos foram comparados aos resultados anteriores por meio de coeficientes de correlação. Identificou-se que não houveram diferenças estatísticas significativas entre os resultados das duas abordagens comparadas.

A relação com este trabalho se dá pelo fato de ele apresentar, de maneira simplificada, um fluxo de trabalho para a realização de testes para avaliação da qualidade de transmissão de vídeo, via streaming. Tal fluxo apresenta atores de um processo (gerente de testes e testador), bem como as ferramentas utilizadas (Quality crowd - framework desenvolvido e a plataforma Amazon Mechanical Turk). Também são representados os artefatos que dão apoio à execução dos testes, bem como às transições destes entre os atores e as ferramentas. Esta representação não pode ser considerada como processo, uma vez que não estão representadas as atividades do fluxo de trabalho. Apesar disso, este trabalho foi considerado por ser o mais próximo a um processo de teste identificado durante a revisão bibliográfica.

Outro aspecto relevante neste trabalho foi o uso da plataforma Amazon Mechanical Turk, executando o fluxo de trabalho em um ambiente real de testes. Por fim, este trabalho está relacionado também por apresentar uma estratégia que utiliza o crowdsourcing para a realização de testes de avaliação da qualidade de transmissão de vídeo.

### 3.3 CROWDSOURCING GUI TESTS

A realização dos testes de interface continua sendo uma atividade que necessita ser realizada por pessoas, em função da dificuldade de um computador determinar se a aparência visual de um software está correta. No trabalho de Dolstra, Vliegendorst e Pouwelse [30] é relatado o uso de crowdsourcing para a realização de testes de interface. O trabalho é baseado na utilização da plataforma Amazon Mechanical Turk (Mturk) para a disponibilização das tarefas à multidão, e a execução do teste é realizada por meio do

uso de Máquinas Virtuais, que são disponibilizadas ao testador em uma página Web. Os testadores devem executar a sequência de passos definida na tarefa e relatar os resultados obtidos. A interação do testador com o software é gravada, e por meio desta gravação os desenvolvedores irão analisar e reproduzir os problemas relatados, facilitando o entendimento do problema a ser solucionado.

O trabalho teve por objetivo explorar dois focos. O primeiro era estabelecer uma estratégia de teste semi-automatizada e contínua, em que, a cada alteração de um software, um sistema de construção contínuo gera uma versão do software e disponibiliza as tarefas na plataforma Mturk para que o projeto seja testado. O segundo objetivo envolvia a realização de estudos de usabilidade, por meio da criação de tarefas em que o trabalhador era desafiado a realizar um determinado objetivo no software, ao invés de executar uma determinada sequência de passos. O uso desta abordagem permite a obtenção de informações qualitativas e quantitativas relacionadas à usabilidade de um software, em que podem ser observadas, por exemplo, as taxas de sucesso e de tempo de conclusão de determinada tarefa. Por meio destas informações é possível então identificar padrões de interação, e o uso de uma plataforma crowdsourcing facilita a mobilização de um grande grupo de pessoas, o que gera dados estatisticamente significativos.

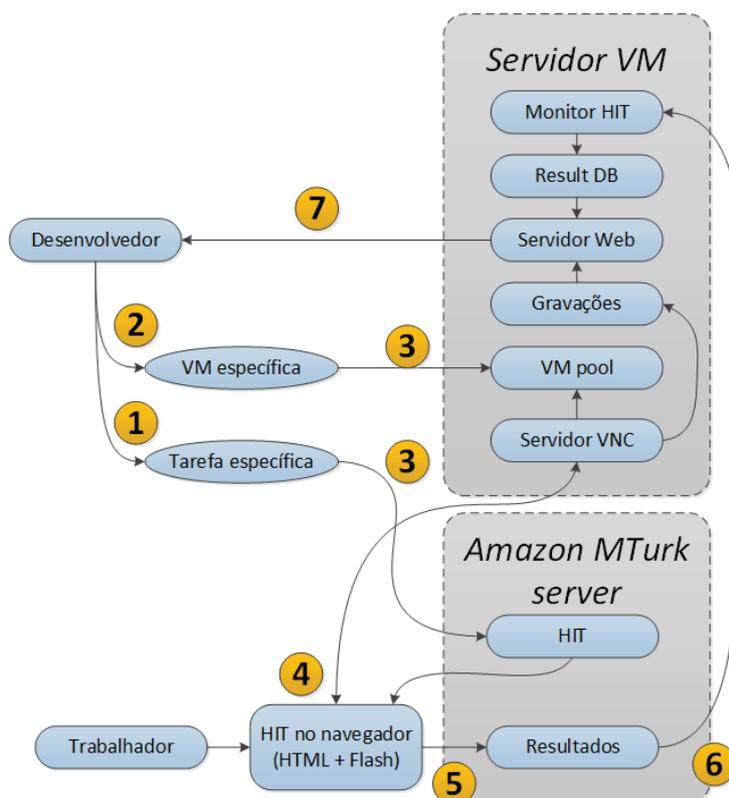


Figura 5. Processo de teste de interfaces por meio do Mturk. Fonte: adaptado de [30, p. 334].

O processo de criação de um teste e de sua execução e coleta dos resultados é descrito a seguir. A Figura 5 ilustra as etapas envolvidas no processo definido pelos autores do trabalho.

1. O desenvolvedor escreve uma descrição da tarefa de teste, que envolve as etapas a serem realizadas pelo testador. A descrição é realizada em um arquivo XML e envolvem a ação a ser executada e a pergunta a ser respondida pelo testador, que pode ser booleana (resposta “sim” ou “não”) ou textual;
2. O desenvolvedor elabora também a descrição da máquina virtual necessária para a execução do teste, que contém informações relacionadas ao software a ser testado e o estado em que a mesma deverá estar para que o testador possa executar o teste;
3. A partir das informações definidas pelo desenvolvedor nos passos 1 e 2, a máquina virtual e a tarefa são criadas e disponibilizadas na plataforma Mturk;
4. O testador acessa a tarefa e aceita a proposta e o acesso à máquina virtual é disponibilizado. O testador realiza o acesso à máquina virtual por meio de seu navegador Web e recebe as informações necessárias para a realização do teste. A cada passo, uma pergunta é disponibilizada ao testador. Quando o testador responde a pergunta, é disponibilizado o passo seguinte. A interação do testador com a máquina virtual é gravada;
5. O testador conclui o teste e as respostas são enviadas para o Mturk;
6. O servidor de máquinas virtuais obtém o resultado da execução do teste, e é realizada uma verificação de aceitação da tarefa, que é então encerrada;
7. Os desenvolvedores passam a ter acesso aos resultados, juntamente com a gravação da realização do teste por parte do testador.

Para a realização dos testes por meio desta abordagem foram elaborados casos de testes com foco no teste das interfaces gráficas para linux KDE e Xfce. Também foram elaborados casos de teste para o software tribler, que é utilizado para compartilhamento de arquivos utilizando a arquitetura *Peer-to-peer* (P2P). Foram cadastradas 51 tarefas de teste na plataforma Mturk, que foram realizadas por 398 testadores de 32 países diferentes, sendo a grande maioria da Índia. Ao todo, foi registrada a execução de 700 testes.

Devido a velocidade de conexão à internet dos trabalhadores indianos ser lenta, houve um efeito no tempo de conclusão das tarefas. Outra questão importante foi o uso da resolução de tela na máquina virtual, que teve que ser reduzida para 800x600 pixels em função de reduzir o deslocamento na tela, o que aumentava o tempo de execução da tarefa.

Como conclusão da realização deste trabalho, os autores consideram que os testadores foram capazes de realizar as tarefas, apesar das limitações de velocidade da conexão

à internet. O uso desta abordagem permitiu a mobilização de um grupo de participantes a um custo menor, em comparativo à abordagens tradicionais de testes de interface e permitiu a participação de testadores externos ao contexto em que o trabalho foi realizado. Adicionalmente, foi considerado como sendo necessário, em alguns casos, descrições mais detalhadas da tarefa que deveria ser realizada pelo testador, que poderia ser realizada por meio de imagens juntamente com a descrição textual, além da qualificação dos desenvolvedores e dos testadores, como forma de se obter melhores resultados.

Da mesma forma que no trabalho anterior, a relação se dá pelo fato de este trabalho apresentar, de maneira simplificada, um fluxo de trabalho com as etapas realizadas para a realização de testes de interface. Tal fluxo apresenta atores de um processo (desenvolvedor e trabalhador), bem como as ferramentas utilizadas (servidor VM e Amazon Mechanical Turk). Também são representados os artefatos que dão apoio à execução dos testes, bem como a transição de informações entre os atores, ferramentas e artefatos. Esta representação não pode ser considerada como processo, uma vez que não estão representadas as atividades do fluxo de trabalho. Apesar disso, este trabalho foi considerado por ser o mais próximo a um processo de teste identificado durante a revisão bibliográfica.

Outro aspecto relevante neste trabalho foi o uso da plataforma Amazon Mechanical Turk, executando o fluxo de trabalho em um ambiente real de testes, envolvendo a execução do trabalho por testadores reais. Por fim, este trabalho também está relacionado por apresentar uma estratégia de teste que utiliza o crowdsourcing para a realização de testes de interface.

### 3.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Neste capítulo foram descritos trabalhos relacionados com a temática e com o propósito deste trabalho. O primeiro trabalho, descrito na seção 3.1, se relaciona a este trabalho por descrever atributos de qualidade a serem observados no desenvolvimento de plataformas crowdsourcing. Buscou-se, durante o estudo de caso, avaliar se a plataforma crowdsourcing selecionada contemplava os atributos descritos pelos autores.

Os trabalhos descritos nas seções 3.2 e 3.3 se relacionam a este trabalho por apresentarem experiências já realizadas em relação ao uso do crowdsourcing para a execução de testes, especificamente em relação a streaming de vídeo e interface, caracterizando-se como testes que avaliam características não funcionais de um software. Em ambos os casos demonstrou-se a viabilidade do uso do crowdsourcing para a execução dos testes, o que motiva o estudo do uso desta abordagem também em relação ao teste funcional.

Além disso, é importante destacar também que ambos os trabalhos apresentam fluxos que descrevem, mesmo que de maneira simplificada, um “processo” para a execução dos testes. Dada esta constatação, percebeu-se a necessidade da documentação do

processo de teste, o que motivou a busca por técnicas e ferramentas mais adequadas para esta finalidade.

Outro elemento relevante nos trabalhos apresentados foi o uso de softwares reais para a execução dos testes. Nesse sentido, buscou-se, por meio do estudo de caso, executar o processo em um ambiente real de uso em empresas de desenvolvimento de software. Para isso utilizou-se softwares comerciais para a execução do processo de teste, buscando simular o ambiente real de uso do processo com a condução das atividades nas empresas, utilizando plataformas crowdsourcing, recrutando testadores reais, que executaram casos de testes e reportaram falhas efetivamente existentes nos softwares.

Por fim, a relação com poucos trabalhos demonstra a importância de realização de estudos na área de teste de software com crowdsourcing que aliem, além do teste, também um processo, que apresente elementos minimamente necessários para que empresas possam utilizar esta abordagem.

## 4. CROWDSOURCING PROCESS FOR FUNCTIONAL TESTS - CPFT

Este capítulo apresenta o “Crowdsourcing Process for Functional Tests - CPFT”, processo de teste desenvolvido a partir da adaptação do processo de teste proposto por Sommerville [4], e que é a principal contribuição deste trabalho.

Na primeira seção são descritas características da sua construção, bem como técnicas de modelagem e representação de processos utilizadas. Na sequência é descrito o detalhamento do processo, detalhando cada ator envolvido, atividade executada, bem como os artefatos desenvolvidos para apoiar a execução do processo.

### 4.1 CONSTRUÇÃO DO PROCESSO

A construção do processo de teste proposto neste trabalho iniciou-se a partir do estudo do modelo de processo de teste apresentado por Sommerville [4]. Optou-se pelo processo do autor devido ao fato de ser um processo simplificado, composto por poucas atividades de grande relevância. A Figura 6 apresenta o modelo de processo proposto pelo autor, composto por quatro atividades e quatro artefatos.

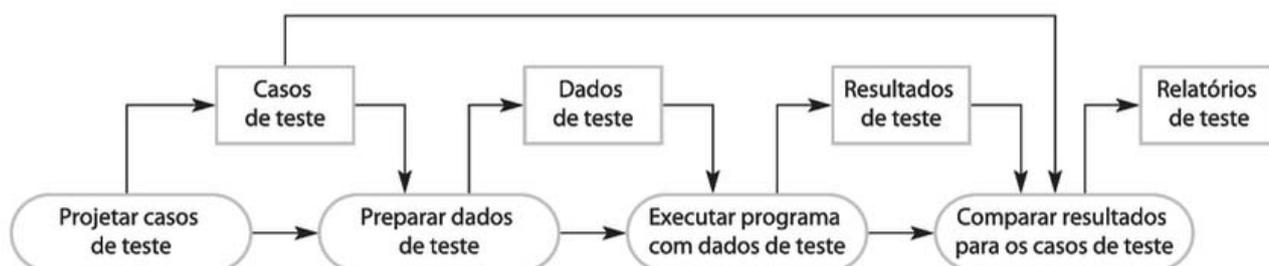


Figura 6. Modelo de processo de teste. Fonte: Sommerville [4, p.147]

O processo possui as seguintes atividades: “Projetar casos de teste”, “Preparar dados de teste”, “Executar programa com dados de teste” e “Comparar os resultados para os casos de teste”. As atividades são descritas a seguir.

A atividade “Projetar casos de teste” é responsável pela elaboração dos casos de teste, que são especificações dos dados de entrada e dos resultados esperados, gerando estes como saída. Na segunda atividade, “Preparar dados de teste”, são preparadas as entradas necessárias, gerando dados de teste que serão utilizados na execução. Na atividade “Executar programa com dados de teste”, os casos de teste projetados são executados diretamente no software, utilizando os dados de teste anteriormente preparados, gerando ao final, resultados de teste. Por fim, na atividade “Comparar resultados para os casos de teste” são comparados de maneira automática os resultados obtidos com os resultados

esperados, obtendo-se assim quais casos de testes que foram aprovados ou reprovados (neste caso indicando falhas no software). Os resultados obtidos são então especificados em um relatório de teste.

O processo de teste foi então executado durante a etapa de investigação, em uma empresa de desenvolvimento de software da cidade de Passo Fundo. Para isto, realizou-se o levantamento do processo de desenvolvimento de software que a empresa utilizava, onde identificou-se que esta empresa não realizava testes funcionais de maneira adequada, pois os testes realizados ocorriam de maneira exploratória e não eram realizados quaisquer tipos de documentação ou registro da execução dos testes.

A partir dessas constatações alterou-se o processo de desenvolvimento de software da empresa, incluindo as atividades do processo de Sommerville [4]. Posteriormente o processo foi implantado e executado e as atividades de teste passaram a ser executadas na empresa, possibilitando o controle dos casos de teste e dos resultados.

Em paralelo à execução do processo com as adaptações na empresa, passou-se a investigar quais atividades deste poderiam ser executadas utilizando-se o crowdsourcing. Em um primeiro momento foram analisadas as propostas de Dolstra, Vliegendhart e Pouwelse [30] para a execução de testes de interface e de Keimel et al. [25], para a realização de testes de streaming de vídeo, avaliando quais atividades do processo de teste estes trabalhos apresentavam e de que forma elas eram executadas.

Posteriormente realizou-se uma análise exploratória de algumas plataformas crowdsourcing (99tests, Utest/Applause, Passbrains, Freelancer, Amazon Mechanical Turk). Esta análise consistiu em leitura de documentações disponibilizadas em seus websites, análise de referências bibliográficas que descrevem seu uso, bem como contatos via e-mail e chat com representantes das plataformas, a fim de levantar informações e investigar como estas plataformas poderiam ser integradas a um processo de teste. Durante o levantamento, observou-se também se as plataformas contemplavam os atributos de qualidade descritos por Allahbakhsh et al. [24]. Adicionalmente realizou-se também levantamento de custos, prazos e elementos necessários da empresa para que as plataformas executassem os testes.

Por fim, construiu-se o processo de teste apresentado neste trabalho. Utilizou-se a notação BPMN para a representação visual do processo. A Figura 7 apresenta o processo proposto. As atividades destacadas em cinza foram incluídas do processo de Sommerville [4] e são a base da execução do processo: “Projetar casos de teste” é correlata à atividade “3. Elaborar os casos de teste”, “Preparar dados de teste” é relacionada à atividade “4. Configurar ambiente de teste” e as atividades “Executar programa com dados de teste” e “Comparar resultados para os casos de teste” estão relacionadas com a atividade “6. Executar tarefa de teste”.

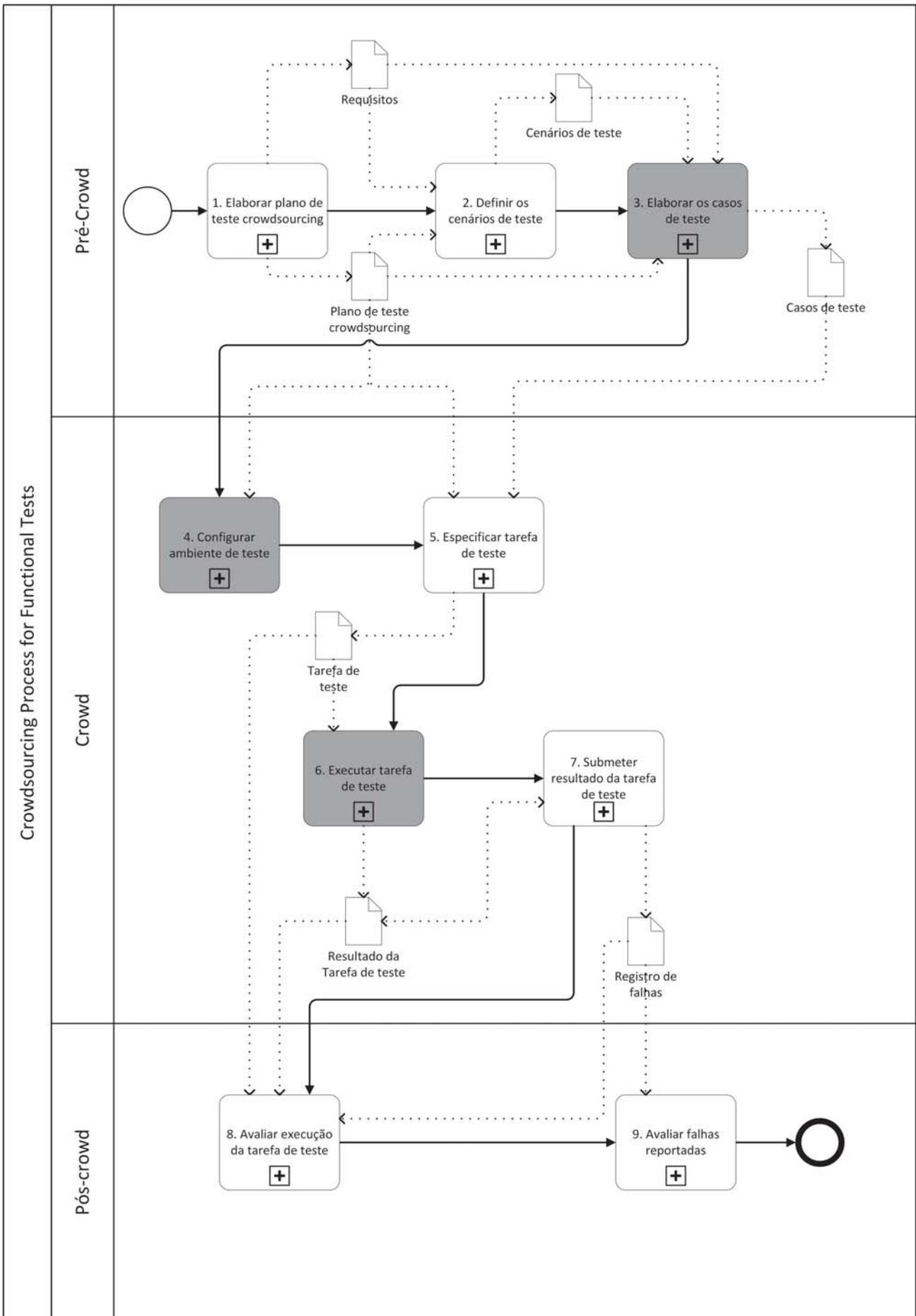


Figura 7. Diagrama BPMN do processo. Fonte: elaborada pelo autor.

Uma das preocupações levadas em consideração na elaboração do processo foi a necessidade de que o mesmo fosse adaptável em relação à documentação. Apesar do processo apresentar artefatos e documentos que são obrigatórios, o uso dos modelos de documentações disponibilizados é opcional, ou seja, os modelos são sugestões. Tais documentos devem ser adaptados aos documentos que a empresa já utiliza (incluindo os elementos presentes nos modelos e que não são contemplados) ou então podem ser substituídos por ferramentas, como por exemplo, para a gestão dos casos de teste ou registro de falhas. No caso da empresa que não possui ou utiliza determinado artefato existente no processo, esta deve passar a utilizá-lo, de acordo com os modelos disponibilizados. Além disso, os artefatos do processo também estão acompanhados de exemplos preenchidos.

Em relação ao processo buscou-se que fosse adaptável também em relação à integração a outros processos já em uso. O processo pode ser adaptado ao processo de desenvolvimento de software utilizado pela empresa, permitindo que ele seja integrado e alinhado com as atividades realizadas. O processo também pode ser executado isoladamente, caso a empresa não utilize um processo de desenvolvimento de software.

Outra preocupação foi a não adoção de uma plataforma crowdsourcing específica para o uso do processo. Essa premissa é importante para que o processo não se restrinja ao uso de apenas uma plataforma crowdsourcing, possibilitando que a empresa decida qual a plataforma que melhor se adéqua às suas necessidades.

O processo apresenta os elementos mínimos para que seja possível seu uso, propondo nove atividades a serem executadas:

1. Elaborar plano de teste crowdsourcing;
2. Definir os cenários de teste;
3. Elaborar os casos de teste;
4. Configurar ambiente de teste;
5. Especificar tarefa de teste;
6. Executar tarefa de teste;
7. Submeter resultado da tarefa de teste;
8. Avaliar execução da tarefa de teste;
9. Avaliar falhas reportadas.

As atividades “5. Especificar tarefa de teste”, “7. Submeter resultado da tarefa de teste” e “8. Avaliar execução da tarefa de teste” foram incluídas no processo em função do uso do crowdsourcing como estratégia para execução dos testes funcionais. As atividades “1. Elaborar plano de teste crowdsourcing” e “9. Avaliar falhas reportadas”, por sua vez,

foram incluídas e/ou adaptadas do processo de Sommerville [4] devido à necessidade de planejamento e controle do teste, bem como do resultado obtido. Tarefas de avaliação dos resultados foram incluídas a partir da recomendação do trabalho de Chen e Luo [31] onde os autores consideram necessária a realização de inspeções em resultados de testes obtidos, em virtude das habilidades irregulares dos testadores e também das possíveis fraudes na realização das tarefas crowdsourcing.

Além da representação do processo por meio de um diagrama BPMN, foi necessária a elaboração de sua documentação, descrevendo cada fase, atividade, tarefa, artefato e ator do processo. Para a organização da documentação do processo utilizou-se o software EPF (Eclipse Process Framework) Composer, que é mantido pela Eclipse Foundation. O software categoriza e organiza os elementos do processo, realizando links entre as atividades, atores e artefatos. O EPF Composer permite também que sejam inseridos diagramas do processo, bem como a inserção de links diretamente no diagrama, permitindo o acesso às informações dos elementos visualmente representados.

Como produto final, o EPF Composer gera, para cada elemento do processo, páginas em formato HTML, que podem ser visualizadas localmente ou disponibilizadas em uma página web. A documentação do processo foi disponibilizada por meio da url “<http://www.softwarecrowdsourcing.com.br/cpft/>”, possibilitando o acesso e o uso da documentação do processo nas empresas durante a condução do estudo de caso.

A Figura 8 apresenta a organização dos elementos do processo, realizada pelo EPF Composer. Nesta organização, foram definidas as seguintes seções para a representação do processo:

- **Fases do processo:** descreve as fases do processo: Pré-crowd, crowd e pós-crowd, bem como relaciona cada fase às suas respectivas atividades;
- **Atores:** relaciona cada um dos atores do processo, bem como relaciona os mesmos às atividades do processo;
- **Artefatos do processo:** relaciona os artefatos do processo, definidos para dar apoio à execução das atividades;
- **Modelos:** apresenta modelos desenvolvidos para os artefatos do processo;
- **Exemplos:** apresenta exemplos de utilização dos artefatos definidos para o processo.

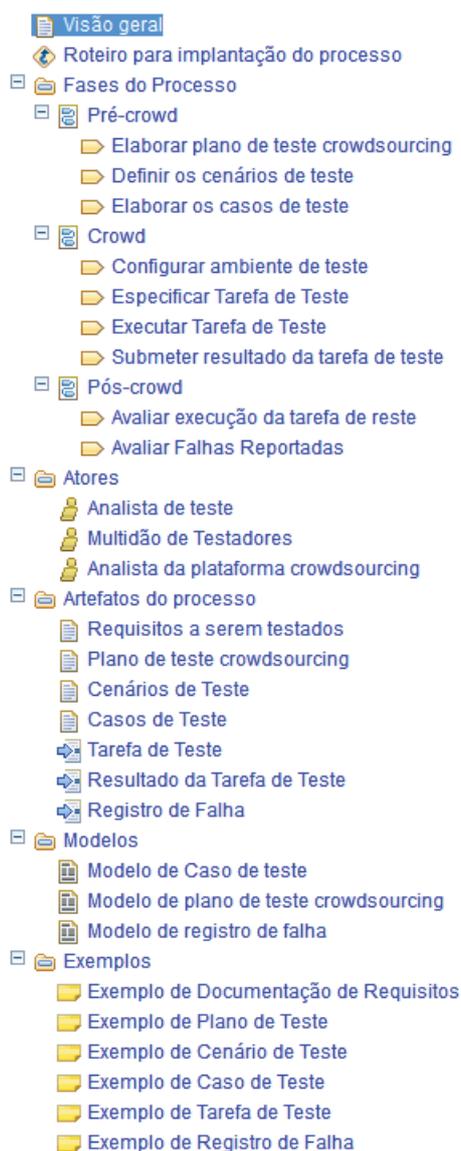


Figura 8. Organização dos elementos na documentação do processo. Fonte: elaborada pelo autor.

A Figura 9 apresenta, a título de exemplo, a atividade “Elaborar plano de teste crowdsourcing”. A página de descrição de uma atividade é composta pelos seguintes elementos:

- **Título:** Informa o título da atividade;
- **Objetivo:** Informa o objetivo da atividade;
- **Atores:** Relaciona os atores responsáveis pela execução da atividade;
- **Entradas:** Relaciona os artefatos do processo que são entradas para a execução da atividade;
- **Saídas:** Relaciona os artefatos do processo que são saídas para a execução da atividade;

- **Descrição principal:** Descreve e explica a importância da atividade e sua contribuição no processo;
- **Tarefas:** Descreve as ações atômicas que são executadas na atividade.

The screenshot displays the CPFT web application interface. The main content area shows the following details for the task 'Elaborar plano de teste crowdsourcing':

- Task:** Elaborar plano de teste crowdsourcing
- Description:** Envolve o planejamento das atividades de teste a serem realizadas.
- Disciplines:** Pré-crowd
- Purpose:** Organizar e planejar as atividades de teste.
- Relationships Table:**

Roles	Primary Performer:	Additional Performers:
	<ul style="list-style-type: none"> <li>Analista de teste</li> </ul>	
Inputs	<b>Mandatory:</b> <ul style="list-style-type: none"> <li>Requisitos a serem testados</li> </ul>	<b>Optional:</b> <ul style="list-style-type: none"> <li>None</li> </ul>
Outputs	<ul style="list-style-type: none"> <li>Plano de teste crowdsourcing</li> </ul>	
- Main Description:** A elaboração do plano de teste é a primeira atividade do processo. O plano de testes organiza as atividades de

Figura 9. Exemplo de atividade. Fonte: elaborada pelo autor.

A Figura 10 apresenta, a título de exemplo, o ator “Analista de testes”. A página de descrição de um ator é composta pelos seguintes elementos:

- **Nome:** Informa o nome do ator;
- **Relações:** Apresenta as atividades do processo desempenhadas pelo ator, bem como os artefatos do processo que estão sob sua responsabilidade;
- **Descrição principal:** Descreve o ator e sua contribuição no processo;
- **Habilidades:** Descreve as habilidades necessárias ao profissional.

The screenshot displays the CPFT web application interface. At the top left, the 'CPFT' logo is prominent. The navigation menu on the left lists various sections like 'Visão geral', 'Roteiro para implanta', 'Fases do Processo', 'Pré-crowd', 'Crowd', 'Pós-crowd', 'Atores', 'Artefatos do processo', 'Modelos', and 'Exemplos'. The main content area is titled 'Atores > Analista de teste' and 'Role: Analista de testes'. It features a description of the role: 'O Analista de testes é o profissional responsável pela gestão, planejamento, especificação e acompanhamento de resultados dos testes.' Below this is a 'Relationships' diagram showing the 'Analista de teste' role performing activities like 'Avaliar Falhas Reportadas', 'Configurar ambiente de teste', 'Definir os cenários de teste', 'Elaborar os casos de teste', and 'Elaborar plano de teste crowdsourcing', and being responsible for artifacts like 'Casos de Teste', 'Cenários de Teste', 'Plano de teste crowdsourcing', 'Requisitos a serem testados', and 'Tarefa de Teste'. A 'Primary Performs' list at the bottom right includes: 'Avaliar Falhas Reportadas', 'Configurar ambiente de teste', 'Definir os cenários de teste', 'Elaborar os casos de teste', and 'Elaborar plano de teste crowdsourcing'.

Figura 10. Exemplo de ator. Fonte: elaborada pelo autor.

A Figura 11 apresenta, a título de exemplo, o artefato “Plano de teste crowdsourcing”. A página de descrição de um artefato é composta pelos seguintes elementos:

- **Nome:** Informa o nome do artefato;
- **Relações:** Apresenta as relações do artefato com atores. Também são mapeadas as relações do artefato com as atividades, relacionando o artefato como entrada ou saída;
- **Descrição principal:** Descreve o artefato e sua contribuição no processo;
- **Modelos e exemplos:** Relaciona o artefato a seus respectivos modelos e exemplos.

The screenshot displays the CPFT web application interface. The main content area shows the 'Artifact: Plano de teste crowdsourcing' page. The page includes a navigation menu on the left, a breadcrumb trail, and several sections: Purpose, Relationships, and Description. The Relationships section contains a table with columns for Roles, Responsible, Modified By, Input To, and Output From.

**Artefatos do processo > Plano de teste crowdsourcing**  
**Artifact: Plano de teste crowdsourcing**

É um guia para as atividades de teste do processo.

Expand All Sections Collapse All Sections

**Purpose**

Organizar e planejar as atividades de teste.

Back to top

**Relationships**

<b>Roles</b>	<b>Responsible:</b> <ul style="list-style-type: none"><li>Analista de teste</li></ul>	<b>Modified By:</b> <ul style="list-style-type: none"><li>Analista de teste</li></ul>
<b>Tasks</b>	<b>Input To:</b> <ul style="list-style-type: none"><li>Configurar ambiente de teste</li><li>Definir os cenários de teste</li><li>Elaborar os casos de teste</li><li>Especificar Tarefa de Teste</li></ul>	<b>Output From:</b> <ul style="list-style-type: none"><li>Elaborar plano de teste crowdsourcing</li></ul>

Back to top

**Description**

**Main Description** O plano de testes organiza as atividades de teste, levando em conta os objetivos de

Figura 11. Exemplo de artefato. Fonte: elaborada pelo autor.

Para auxiliar as empresas na implantação do processo e na definição de quais documentações devem ser criadas ou adaptadas, elaborou-se um fluxograma que estabelece um roteiro para implantação do processo. A Figura 12 apresenta este fluxograma.

O roteiro de implantação do processo é composto por perguntas, que avaliam se a empresa já realiza atividades do processo de teste, bem como se já utiliza os artefatos de documentação definidos no processo. Quando determinada atividade do processo não é realizada, o fluxograma indica que a empresa deve passar a realizá-la. Em relação aos artefatos do processo, quando a empresa já os utiliza, o fluxograma indica que a empresa deve adaptá-lo, contemplando os elementos disponíveis nos modelos dos artefatos do processo. Caso a empresa não adote determinado artefato, o fluxograma indica que esta deve adotá-lo.

Ao final da execução do roteiro de implantação do processo, a empresa obtém uma relação de elementos que devem ser criados ou adaptados para que o processo de teste possa ser executado.

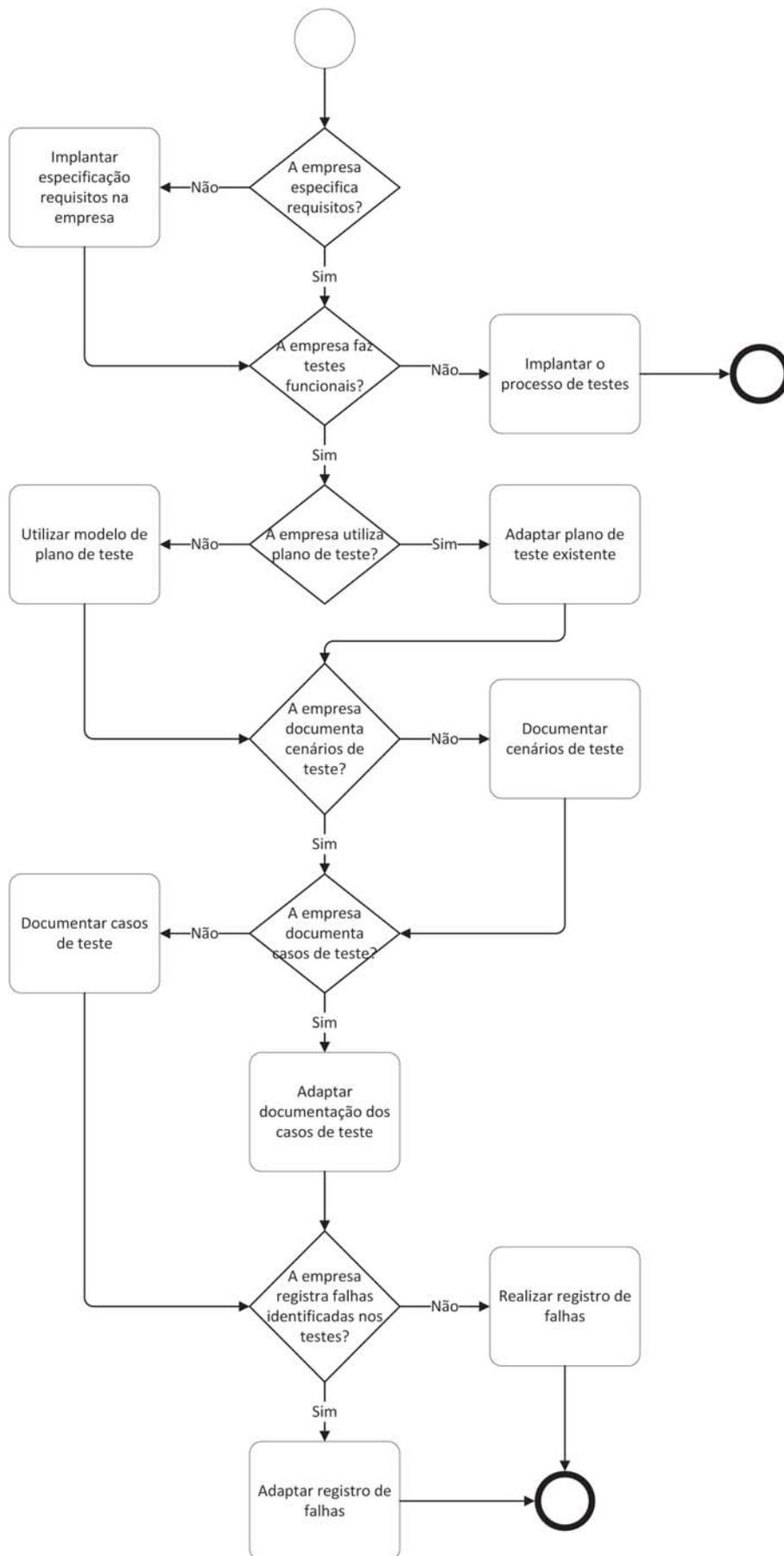


Figura 12. Roteiro para implantação do processo. Fonte: elaborada pelo autor.

As seções a seguir descrevem os elementos do processo: atores, fases, atividades e tarefas e os artefatos.

## 4.2 ATORES DO PROCESSO

Para que as atividades e as tarefas do processo sejam realizadas, foram definidos dois atores obrigatórios e um ator opcional. Os nomes dos atores são apenas sugestões, uma vez que as nomenclaturas variam nas empresas. Além disso, não é necessário que exista um profissional específico na empresa para executar as atividades, podendo estas serem realizadas por um ou mais profissionais como parte de suas atribuições, de acordo com as necessidades e disponibilidades específicas de cada empresa.

### 4.2.1 Analista de testes

O Analista de Testes é o ator responsável por realizar as atividades na empresa, identificando e definindo o que será testado e quais serão os testes necessários, bem como avaliando os resultados obtidos com a execução dos testes. O analista de testes é responsável por:

- Elaborar plano de teste crowdsourcing;
- Definir os cenários de teste;
- Elaborar os casos de teste;
- Configurar ambiente de teste;
- Especificar tarefa de teste;
- Avaliar execução da tarefa de teste;
- Avaliar Falhas Reportadas.

### 4.2.2 Analista da plataforma crowdsourcing

O analista da plataforma crowdsourcing é um ator opcional, responsável por gerenciar a execução dos testes, caso a organização opte pelo uso de uma plataforma crowdsourcing que ofereça o serviço de gestão dos casos de teste, mobilização e pagamento dos testadores e validação dos resultados das tarefas de teste. Quando presente, este ator é responsável por:

- Especificar tarefa de teste;

- Avaliar execução da tarefa de teste.

### 4.2.3 Multidão de testadores

A multidão de testadores é caracterizada por ser uma força de trabalho dispersa geograficamente, ou seja, são testadores dispersos em qualquer local do planeta, diferentemente das metodologias “tradicionais” de teste, em que os testadores são funcionários da empresa. A multidão de testadores é responsável por:

- Executar tarefa de teste;
- Submeter resultado da tarefa de teste.

## 4.3 FASES DO PROCESSO E ATIVIDADES

Esta seção apresenta as fases do processo, bem como as atividades definidas para cada fase, suas tarefas, atores e artefatos relacionados.

De acordo com a Object Management Group (OMG) [32], que mantém a notação BPMN, uma atividade é um trabalho que uma organização executa utilizando processos de negócio, podendo ser atômica (em que passa a ser denominada como tarefa) ou não atômica (neste caso sendo composta por outras atividades ou por tarefas). Cada atividade do processo é composta pelos seguintes elementos:

**Descrição:** Descreve o objetivo da atividade e explica sua importância e contribuição no processo;

**Atores:** Relaciona a atividade com os atores do processo que a executam;

**Entradas:** Artefatos necessários para que a execução da atividade seja realizada;

**Tarefas:** Ações atômicas que são executadas na atividade;

**Saídas:** Artefatos gerados ao final da execução da atividade.

O processo é organizado em três fases: Pré-crowd (em que ocorre o planejamento e elaboração dos testes), Crowd (responsável pela disponibilização e execução das tarefas de teste) e Pós-crowd (envolve a avaliação do resultado reportado para cada tarefa de teste e a avaliação e registro das falhas identificadas). Cada uma das fases é descrita nas Seções a seguir.

#### 4.4 FASE 1 - PRÉ-CROWD

A primeira fase do processo, denominada como **Pré-crowd**, é composta pelas atividades de planejamento da elaboração dos testes, que são realizadas pelo analista de testes. A Figura 13 apresenta as atividades relacionadas nesta fase, bem como as tarefas contidas em cada uma das atividades, que serão descritas a seguir.

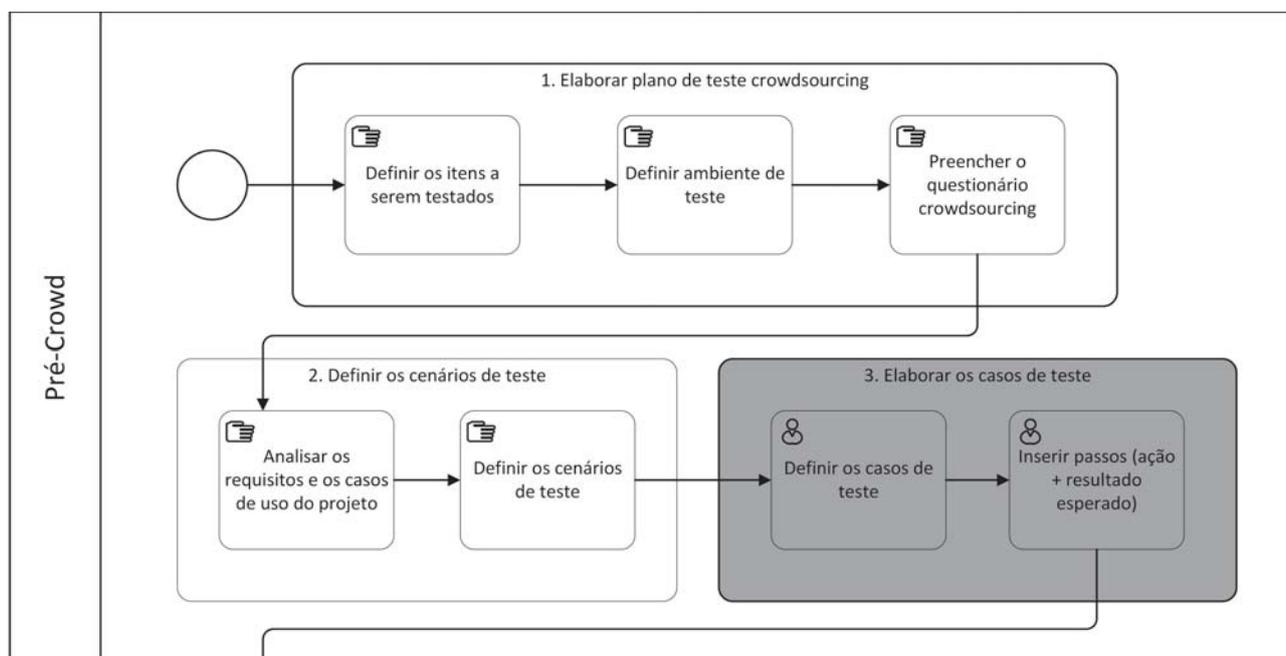


Figura 13. Fase 1 - Pré-crowd. Fonte: elaborada pelo autor.

##### 4.4.1 Atividade 1 - Elaborar plano de teste crowdsourcing

A elaboração do plano de teste é a primeira atividade do processo. O plano organiza e guia as atividades de teste, levando em conta os objetivos, especificando quem faz os testes, porque são realizados, como são conduzidos e quando serão executados.

**Atores:** Analista de testes.

**Entradas:** Documento de requisitos.

**Tarefas:**

- **Definir os itens a serem testados:** Esta tarefa tem por objetivo definir a cobertura dos testes, ou seja, o que será testado. Esta definição deve ser obtida a partir dos requisitos, independente da forma que estejam representados (documento de requisitos, casos de uso da UML, dentre outras formas de representação de requisitos);
- **Definir ambiente de teste:** Esta tarefa tem como objetivo definir e relacionar os recursos necessários para a execução do plano de teste, envolvendo a forma que

o software será disponibilizado para teste, a massa de dados necessária para a execução dos testes e as configurações necessárias no ambiente de teste. Também é nesta etapa que é definida a plataforma crowdsourcing que será utilizada para a execução dos testes;

- **Preencher o questionário crowdsourcing:** Esta tarefa tem por objetivo realizar o preenchimento da seção do plano de teste referente ao questionário crowdsourcing. Este questionário compila algumas questões que deverão ser observadas, planejadas e respondidas possibilitando o uso do crowdsourcing para a realização dos testes.

**Saídas:** Plano de teste crowdsourcing.

#### 4.4.2 Atividade 2 - Definir os cenários de teste

Esta atividade envolve o levantamento dos cenários em que serão realizados os testes. Tal levantamento é realizado a partir dos requisitos, que podem ser transformados e representados por meio dos casos de uso. Cada caso de uso pode ser mapeado como um cenário de teste. Como forma de descrever os cenários, podem ser utilizados os diagramas de atividades, que apresentam as atividades que são realizadas em cada cenário de teste.

**Atores:** Analista de testes.

**Entradas:**

- Documento de requisitos;
- Plano de teste crowdsourcing.

**Tarefas:**

- **Analisar os requisitos e os casos de uso do projeto:** Esta tarefa tem por objetivo analisar os requisitos e os casos de uso do projeto, a fim de identificar os cenários de teste. Recomenda-se a utilização de casos de uso da UML para a definição dos cenários de uso, pois descrevem os requisitos funcionais, identificam o valor que o cliente espera obter por meio do software e representam como o software será utilizado;
- **Definir os cenários de teste:** A partir da análise dos casos de uso, realizada na tarefa anterior, são derivados os cenários de teste para os requisitos definidos no plano de teste. Opcionalmente, a partir de cada cenário de teste, podem ser elaborados diagramas de atividades da UML para descrever os cenários de teste que serão foco da elaboração dos casos de teste, identificando os fluxos a serem testados em cada cenário.

**Saídas:** Cenários de teste.

### 4.4.3 Atividade 3 - Elaborar os casos de teste

A elaboração dos casos de teste envolve inicialmente a análise dos requisitos e dos cenários de uso levantados, buscando identificar os elementos e comportamentos a serem testados. Após esta análise são definidos quais casos de teste serão elaborados. Para cada caso de teste são descritos passos, compostos por uma ação e um resultado esperado.

**Atores:** Analista de testes.

**Entradas:**

- Documento de requisitos;

- Plano de teste crowdsourcing;
- Cenários de teste.

**Tarefas:**

- **Definir os casos de teste:** Esta tarefa tem por foco, a partir da análise dos requisitos e dos cenários de teste levantados, identificar os casos de teste que deverão ser elaborados;

- **Inserir passos (ação + resultado esperado):** Para cada caso de teste são definidos os passos de execução, que são compostos por uma ação a ser executada e um resultado que é esperado a partir da execução da ação.

**Saídas:** Casos de teste.

## 4.5 FASE 2 - CROWD

A segunda fase, denominada de **Crowd**, é a fase responsável pela disponibilização e execução das tarefas de teste. As atividades desta fase são realizadas pelo analista de testes e pelos testadores geograficamente dispersos. A Figura 14 apresenta as atividades relacionadas nesta fase, bem como as tarefas contidas em cada uma das atividades, que serão descritas a seguir.

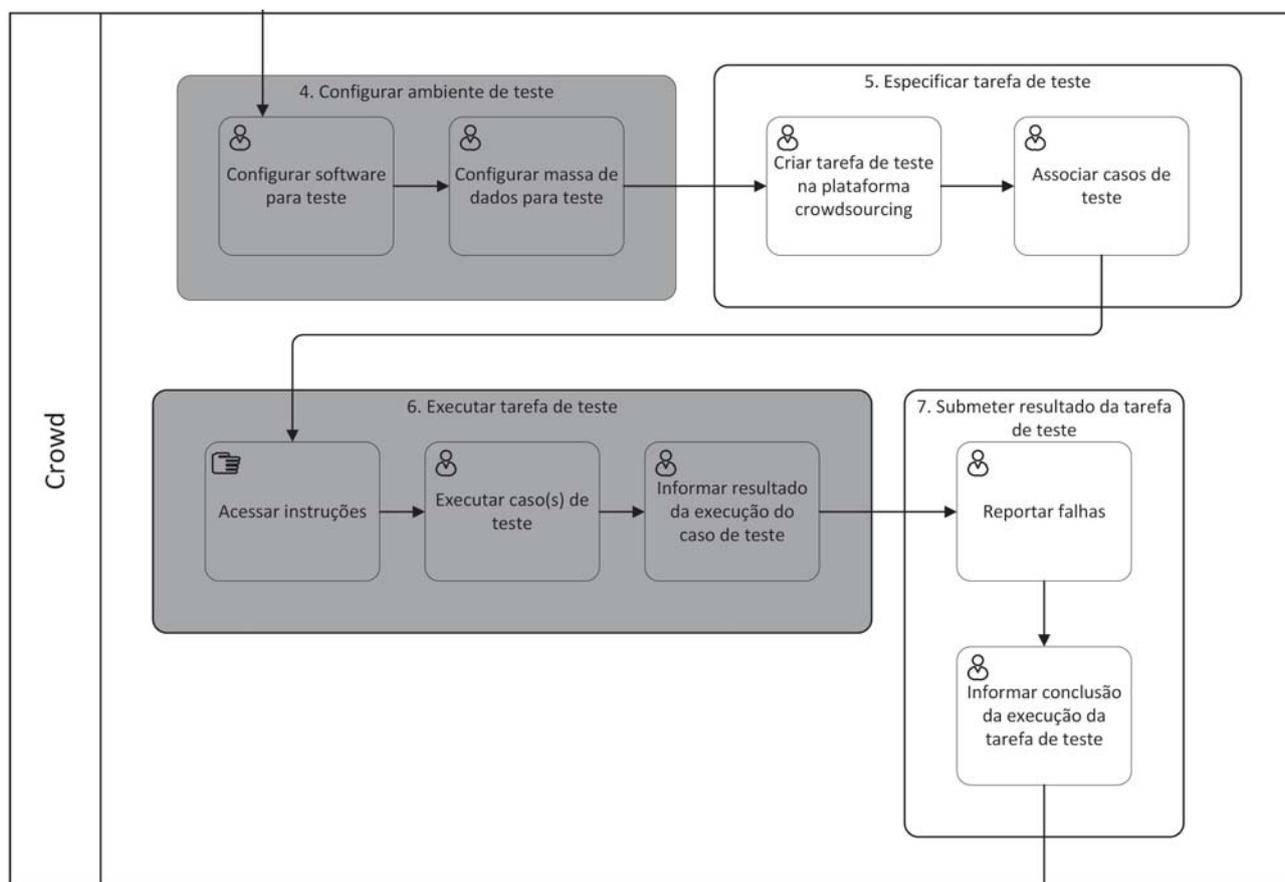


Figura 14. Fase 2 - Crowd. Fonte: elaborada pelo autor.

#### 4.5.1 Atividade 4 - Configurar ambiente de teste

Esta atividade objetiva configurar o ambiente de teste para a execução dos testes, de acordo com o definido no plano de teste. Envolve as tarefas de configuração do software para teste e da massa de dados necessária para a execução dos testes.

**Atores:** Analista de testes.

**Entradas:** Plano de teste crowdsourcing.

**Tarefas:**

- **Configurar software para teste:** Nesta tarefa é configurado o software para a execução dos testes, envolvendo a parametrização, permissões de usuário e a forma de disponibilização do mesmo para acesso do testador, conforme definido no plano de teste;
- **Configurar massa de dados para teste:** Nesta tarefa é configurada a massa de dados necessária para a execução dos testes.

**Saídas:** Esta atividade não gera saídas.

#### 4.5.2 Atividade 5 - Especificar Tarefa de Teste

Nesta atividade ocorre a especificação da tarefa de teste na plataforma crowdsourcing definida no plano de teste, incluindo a descrição em detalhes da tarefa. Esta atividade pode ser realizada por um analista da empresa, caso opte-se por uma plataforma que permita a gestão pela empresa ou; conduzida por um analista da plataforma crowdsourcing, caso a organização decida pelo uso de uma plataforma que disponibilize este profissional para gestão dos casos de teste, mobilização e pagamento dos testadores e validação dos resultados dos testes.

**Atores:** A atividade é realizada por um dos seguintes atores:

- Analista de testes;
- Analista da plataforma crowdsourcing.

**Entradas:**

- Plano de teste crowdsourcing;
- Casos de teste.

**Tarefas:**

- **Criar tarefa de teste na plataforma crowdsourcing:** Nesta tarefa é criada a tarefa de teste na plataforma crowdsourcing, bem como são descritas as informações relevantes para a tarefa de teste, obtidas a partir do plano de teste e dos casos de teste especificados;
- **Associar casos de teste:** Esta tarefa envolve a associação dos casos de teste à tarefa de teste.

**Saídas:** Tarefa de teste.

#### 4.5.3 Atividade 6 - Executar Tarefa de Teste

A atividade “Executar tarefa de teste” envolve o acesso às instruções da tarefa, o acesso ao software a ser testado, a execução do(s) caso(s) de teste associado(s) e a reportação do resultado da tarefa do teste, realizada por um testador recrutado por meio da plataforma crowdsourcing.

**Atores:** Multidão de testadores.

**Entradas:** Tarefa de teste.

**Tarefas:**

- **Acessar instruções:** Nesta tarefa o testador acessa as instruções da tarefa de teste e o(s) caso(s) de teste associados;

- **Executar caso(s) de teste:** Nesta tarefa o testador executa os passos definidos no(s) caso(s) de teste, comparando o resultado obtido em cada passo, com o resultado esperado;
- **Informar resultado da execução do caso de teste:** Nesta tarefa o testador informa o resultado da execução de cada caso de teste, informando se o mesmo passou ou não passou e anexando evidências comprobatórias do resultado do teste (quando solicitado).

**Saídas:** Resultado da tarefa de teste.

#### 4.5.4 Atividade 7 - Submeter resultado da tarefa de teste

Esta atividade tem como objetivo reportar as falhas localizadas pelo testador durante a execução da tarefa de teste, envolvendo a criação do registro da falha, a descrição da falha (incluindo os passos para a sua reprodução) e a submissão da falha. Posteriormente, a conclusão da tarefa é informada ao analista de testes, para que a execução do processo prossiga.

**Atores:** Multidão de testadores.

**Entradas:** Resultado da tarefa de teste.

**Tarefas:**

- **Reportar falhas:** Nesta tarefa o testador preenche o registro de falhas com as falhas identificadas durante a execução dos casos de teste. Para cada falha são descritos detalhes que permitam a sua identificação, bem como os passos para que a mesma possa ser reproduzida;
- **Informar conclusão da execução da tarefa de teste:** Nesta tarefa o testador informa ao analista de testes que concluiu a execução da tarefa de teste, reportando os resultados dos casos de teste executados e o registro com as falhas identificadas;

**Saídas:**

- Registro de falha;
- Resultado da tarefa de teste.

#### 4.6 FASE 3 - PÓS-CROWD

O **Pós-crowd** é a última fase do processo. Nesta fase ocorre a avaliação da execução da tarefa de teste, bem como a avaliação e registro das falhas reportadas pelos testadores. A Figura 15 apresenta as atividades relacionadas nesta fase, bem como as tarefas contidas em cada uma das atividades, que serão descritas a seguir.

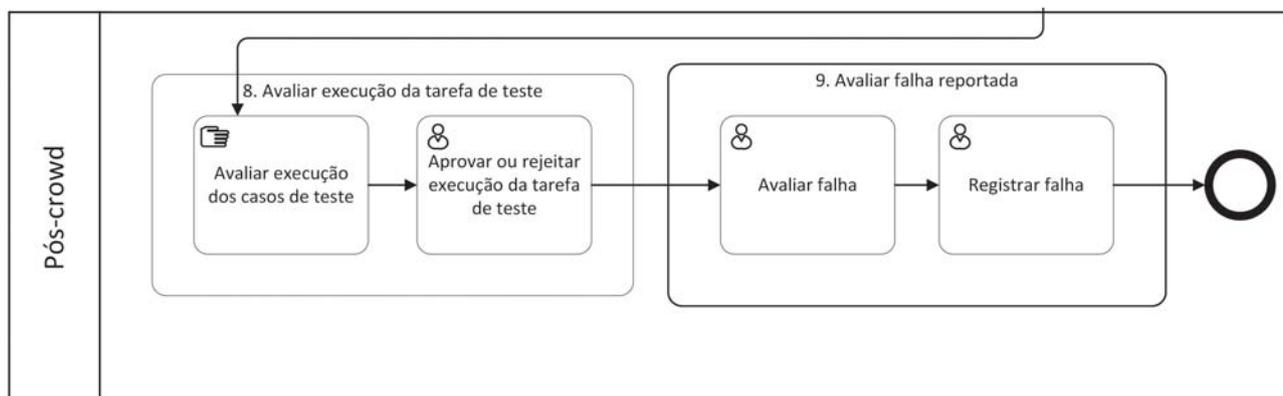


Figura 15. Fase 3 - Pós-crowd. Fonte: elaborada pelo autor.

#### 4.6.1 Atividade 8 - Avaliar execução da tarefa de teste

Nesta atividade é avaliado o resultado da execução da tarefa de teste, aprovando-o ou rejeitando-o. O testador recebe o pagamento pela execução da tarefa apenas se o resultado da execução da tarefa de teste for aprovado.

**Atores:** A atividade é realizada por um dos seguintes atores:

- Analista de testes;
- Analista da plataforma crowdsourcing.

**Entradas:**

- Registro de falha;
- Resultado da tarefa de teste.

**Tarefas:**

- **Avaliar a execução dos casos de teste:** Nesta tarefa o analista de teste (da empresa ou da plataforma, caso opte-se por uma plataforma que disponibilize o profissional) avalia o resultado recebido referente à tarefa de teste, por meio do resultado informado em cada caso de teste e também as informações disponíveis no registro de falhas;
- **Aprovar ou rejeitar a execução da tarefa de teste:** Nesta tarefa o analista informa se aprova ou recusa o resultado da execução da tarefa de teste. O testador recebe o pagamento pela execução da tarefa apenas se o resultado da execução da tarefa de teste for aprovado.

**Saídas:** Esta atividade não gera saídas.

#### 4.6.2 Atividade 9 - Avaliar Falhas Reportadas

Nesta Atividade o analista de testes avalia as falhas reportadas pelo testador, identificadas durante a execução da tarefa de teste. Caso determinada falha realmente exista, o analista de testes deve registrá-la e providenciar a sua correção.

**Atores:** Analista de testes.

**Entradas:** Registro de falha.

**Tarefas:**

- **Avaliar falha:** Nesta tarefa o analista de testes avalia (aprova ou reprova) a falha registrada pelo testador referente a execução da tarefa de teste.
- **Registrar falha:** Caso a falha exista, o testador efetua o registro da falha, utilizando uma ferramenta adequada para tal.

**Saídas:** Esta atividade não gera saídas.

### 4.7 ARTEFATOS DO PROCESSO

Juntamente com a conclusão da elaboração do processo, foram elaborados modelos dos artefatos, bem como exemplos de utilização, para dar suporte à execução das atividades pelas empresas. A seguir são descritos os artefatos desenvolvidos ou adaptados para dar suporte às atividades e tarefas do processo.

#### 4.7.1 Documento de requisitos

São as declarações de serviços que o software deve fornecer, de como este deve reagir a entradas específicas e de como deve se comportar em determinadas situações. O processo não define um modelo específico para a especificação dos requisitos a serem testados, porém sugere que seja utilizado o diagrama de casos de uso da UML, juntamente com a descrição textual de cada caso de uso. O diagrama de casos de uso apresenta de maneira facilmente compreensível os requisitos de um software, identificando as funcionalidades que o mesmo irá disponibilizar [33].

#### 4.7.2 Plano de teste crowdsourcing

O plano de testes organiza as atividades de teste, levando em conta seus objetivos, sendo um guia para a atividade de teste, explicando quem faz os testes, porque são realizados, como são conduzidos e quando serão executados.

O modelo de plano de teste disponibilizado foi elaborado tendo como base a norma IEEE 829-2008 [34], que estabelece padrões para a documentação de testes de software. Adicionalmente, foi incluído no plano de teste o questionário crowdsourcing, devido as especificidades da abordagem de teste crowdsourcing. O questionário compila questões que devem ser observadas, planejadas e respondidas possibilitando o uso de plataformas crowdsourcing para a execução dos testes.

### **4.7.3 Cenários de teste**

Cada requisito funcional de um software pode ser mapeado em um ou mais cenários de teste. Os cenários são importantes, pois a partir deles são derivados os casos de teste. Como forma de descrever os cenários, é sugerido no processo o uso de diagramas de atividades da UML, que descrevem os passos a serem percorridos para a realização de uma atividade no software [33].

### **4.7.4 Caso de teste**

O caso de teste é um conjunto de passos a serem executados para realizar determinado teste em um software. É elaborado com foco em identificar falhas no software, a partir de situações que exercitam suas estruturas. Também é utilizado para verificar se os requisitos especificados foram atendidos. O modelo de caso de teste disponibilizado também foi elaborado tendo como base a norma IEEE 829-2008 [34].

### **4.7.5 Tarefa de teste**

A tarefa de teste é um artefato elaborado para descrever as atividades a serem realizadas pelo testador. Envolve as questões levantadas no questionário crowdsourcing durante a elaboração do plano de teste, bem como os casos de teste desenvolvidos. É por meio da tarefa de teste que os testadores se candidatam à execução dos testes, são selecionados, executam os testes e, por fim, reportam os resultados obtidos, bem como as falhas identificadas.

### **4.7.6 Resultado da tarefa de teste**

O resultado da tarefa de teste é o produto final da realização da tarefa de teste, e é composto pelo registro da execução de cada caso de teste associado. Por meio do resultado da tarefa de teste o analista de testes define se aprova ou não a execução da

tarefa de teste. Caso o resultado não seja aprovado, o testador não recebe a remuneração referente à execução da tarefa de teste.

#### 4.7.7 Registro de falha

O registro de falha é um documento que descreve uma falha identificada durante a execução de um caso de teste. O modelo de registro de falha disponibilizado também foi elaborado tendo como base a norma IEEE 829-2008 [34].

### 4.8 CONSIDERAÇÕES SOBRE O CAPÍTULO

Este capítulo apresentou o CPFT - “Crowdsourcing Process for Functional Tests”, bem como aspectos da sua concepção, a partir do processo de teste de Sommerville [4]. Posteriormente, cada elemento do processo é descrito: atores, fases (pré-crowd, crowd e pós-crowd), atividades, tarefas e artefatos.

Para a elaboração do processo, foi necessária inicialmente a imersão em uma empresa de desenvolvimento de software, a fim de identificar as dificuldades que as empresas enfrentam para implantar um processo, ou modificar um processo existente. Também identificou-se a dificuldade enfrentada por pequenas empresas em relação a demanda de testes, relacionada ao custo de um testador, bem como a variabilidade da demanda de trabalho deste profissional.

Quanto ao processo, buscou-se identificar os elementos mínimos necessários para que as atividades fossem executadas. Além disso, o processo é adaptável quanto à documentação e às ferramentas e plataformas utilizadas. Este aspecto é relevante para que as empresas possam integrar o processo ao processo de desenvolvimento de software que já utilizam, adaptando seus artefatos e ferramentas em uso.

Para a avaliação do processo, descrita no capítulo a seguir, utilizou-se a técnica de estudo de caso, composto por um único contexto e duas unidades de análise (empresas de desenvolvimento de software), nas quais o processo foi executado e foram coletadas evidências para a elaboração do relatório do estudo de caso. Por fim, cada uma das proposições do estudo de caso é avaliada, de forma a confirmar ou refutar as hipóteses anteriormente levantadas.

## 5. ESTUDO DE CASO

Este capítulo apresenta o relatório da realização do estudo de caso, como estratégia de avaliação do processo de teste proposto neste trabalho.

A questão de estudo definida para o estudo de caso é: “Como o processo de teste se comporta ao ser utilizado em uma pequena empresa de desenvolvimento de software?”. A partir desta questão elaborou-se o projeto de pesquisa do estudo de caso (Apêndice A), que aponta 15 proposições a serem avaliadas. Cada uma destas proposições, ao final da coleta de dados, será confirmada, refutada ou permanecerá inconclusiva, caso a condução do estudo de caso não apresente elementos suficientes para isto. Além disso, estabeleceu-se no projeto de pesquisa critérios para a seleção das unidades de análise do estudo de caso.

Posteriormente, a partir do projeto de pesquisa elaborou-se o protocolo para o estudo de caso (Apêndice B). Por meio do protocolo são descritos os procedimentos de campo realizados, princípios de proteção da identidade dos participantes, questões de estudo de caso, que guiarão a investigação e a coleta de dados e um breve guia com a organização do relatório do estudo de caso.

### 5.1 APRESENTAÇÃO DO ESTUDO PILOTO

Esta seção apresenta a condução do estudo piloto, realizado antes da execução do estudo de caso. O estudo piloto foi realizado com o objetivo de verificar se o processo estava adequadamente organizado e documentado, bem como para refinar os procedimentos definidos no projeto e no protocolo do estudo de caso. A condução do estudo piloto foi realizada na empresa A, descrita na Subseção 5.2.2.

Yin [11] afirma que o estudo piloto ajuda a refinar os planos de coleta de dados em relação ao conteúdo e aos procedimentos. Nesse sentido, realizou-se o estudo piloto para verificar se o processo era adequadamente conduzido, se as atividades e tarefas estavam representadas no processo e se não haviam sido omitidas tarefas. Além disso, foi verificado se as atividades representadas no processo permitiam o uso de qualquer plataforma crowd-sourcing ou se haviam atividades que restringiam seu uso a uma determinada plataforma específica. Também foi verificado se a documentação de apoio do processo e do estudo de caso foi elaborada adequadamente, tanto em relação ao conteúdo quanto à relevância das atividades planejadas no projeto e no protocolo do estudo de caso.

Os procedimentos de campo planejados no protocolo para o estudo de caso piloto foram realizados em 19 dias. Neste período ocorreram visitas à empresa para execução das atividades, alocando períodos de no máximo uma hora e meia de trabalho, a fim de não

prejudicar as demais atividades em andamento na empresa. Posteriormente, realizaram-se ajustes no processo e em sua documentação, no projeto e no protocolo do estudo de caso, para que o mesmo pudesse ser conduzido. A seguir é descrita a condução do estudo piloto realizado na empresa A.

#### Dia 1

O procedimento de campo na empresa A foi iniciado com a reunião de apresentação e planejamento das atividades. Inicialmente foi apresentado o objetivo do trabalho que seria realizado na empresa, bem como o planejamento por meio dos procedimentos de campo que seriam realizados. Explanou-se que a intervenção seria realizada em dois momentos: o primeiro como estudo piloto e posteriormente como unidade de análise de um estudo de caso.

Na sequência realizou-se o levantamento de informações da empresa, por meio de uma entrevista guiada por um roteiro semi-estruturado, composto por perguntas definidas anteriormente e por questionamentos que surgissem no decorrer da entrevista (Apêndice C). Com base nesta entrevista foi possível estabelecer um panorama geral da empresa, suas características e particularidades. A representação do processo de desenvolvimento de software que a empresa utiliza foi disponibilizada por meio de um diagrama BPMN, acompanhado de uma breve descrição das atividades da empresa.

#### Dia 2

No segundo dia de atividades na empresa foi apresentado o processo de teste, suas atividades, atores e artefatos, utilizando a documentação desenvolvida para suporte do mesmo. Posteriormente, realizou-se a integração do processo de teste com o processo em uso na empresa (levantado no dia anterior), com auxílio do “Roteiro para implantação do processo”. Por fim, elaborou-se um diagrama BPMN representando a implantação do processo de teste proposto juntamente com o processo que a empresa utiliza. Este diagrama serviu posteriormente como base para a execução do processo.

#### Dia 3

No terceiro dia de atividades na empresa definiu-se o software que seria utilizado para a execução do estudo piloto e foram obtidos os recursos necessários para o início da execução do processo (acesso ao software a ser testado e também ao software de gestão de projetos). Optou-se pela aplicação de testes em um módulo do software ERP, responsável pelo cadastro de clientes e fornecedores.

Esta escolha levou em consideração a disponibilidade do software pela empresa, bem como pela facilidade de acesso para o testador. Como o software disponibilizado pela empresa é do tipo “aplicação web”, não foi necessário que o testador instalasse o software em seu computador para que a execução dos testes fosse realizada.

Dias 4, 5 e 6

No quarto, quinto e sexto dia de atividades, foi executada a atividade do processo “1. Elaborar plano de teste crowdsourcing”. Utilizou-se o modelo de plano de teste disponibilizado pelo processo, uma vez que a empresa não possui documento padronizado próprio para adaptação. A documentação dos requisitos, artefato de entrada desta atividade, teve que ser elaborada, pois a empresa não adota o uso de documentação de requisitos. Utilizou-se a abordagem de casos de uso para representar os requisitos do software.

Dia 7

No sétimo dia de atividades, foi executada a atividade do processo “2. Definir os cenários de teste”. A partir dos casos de uso, foi possível definir os cenários de teste. Foram elencados os seguintes cenários: Cadastro e alteração de dados básicos do cadastro de cliente. Adicionalmente, foram elaborados diagramas de atividades para a documentação dos possíveis fluxos em cada cenário.

Dias 8, 9 e 10

No oitavo, nono e décimo dia de atividades na empresa foi executada a atividade do processo: “3. Elaborar os casos de teste”. Para a elaboração dos casos de teste, inicialmente utilizou-se o modelo de caso de teste disponibilizado no processo. Para o estudo piloto, foram elaborados 20 casos de teste para os cenários de teste elencados, utilizando os critérios de particionamento em classes de equivalência, análise do valor limite, teste funcional sistemático e error guessing. Posteriormente, os casos de teste foram priorizados, traduzidos para o idioma inglês e inseridos na plataforma testlink<sup>1</sup>, utilizada para o registro da execução dos casos de teste. O uso da plataforma testlink foi necessário, uma vez que para o uso do Microsoft Test Manager (plataforma que a empresa A utiliza para gestão dos casos de teste) é necessário instalá-lo no computador do testador, bem como configurar o acesso ao servidor da empresa, o que não foi possível, uma vez que o acesso só é possível na rede local da empresa.

---

<sup>1</sup>Disponível em <<http://testlink.org/>>.

Dias 11 e 12

No décimo primeiro e décimo segundo dia de atividades na empresa foi realizada a atividade “4. Configurar ambiente de teste”.

Em relação à configuração do ambiente de teste, foi necessário instalar o software a ser testado em um servidor exclusivamente para a execução de testes. Também foi criado um usuário fictício no software, bem como foram limitadas as suas permissões de acesso apenas ao necessário para a execução dos casos de teste. Devido ao software-alvo do processo ser uma aplicação web, o mesmo foi disponibilizado ao testador por meio do endereço (URL), informado em cada caso de teste.

Dias 13 e 14

No décimo terceiro e décimo quarto dia de atividades foi executada a atividade “5. Especificar tarefa de teste”. A especificação da tarefa de teste foi realizada a partir das informações do questionário crowdsourcing, disponíveis no plano de teste.

A tarefa de teste possui a característica descrita por [23, 24], que consideram que em algumas tarefas crowdsourcing os trabalhadores necessitam possuir conhecimentos e experiência específicos para realizá-las. Considerou-se esta questão na elaboração do plano de teste crowdsourcing, durante a realização da pesquisa de quais plataformas poderiam ser utilizadas para a execução do processo. Para cada plataforma, foram identificadas características e peculiaridades, bem como o custo. Foram consideradas nesta escolha as seguintes plataformas: Utest/Applause, Passbrains, Testbirds, Crowdtest, Amazon Mechanical Turk e Freelancer.

Inicialmente levantaram-se as informações em relação às plataformas crowdsourcing específicas para a execução de testes (Utest/Applause, Passbrains, Testbirds e Crowdtest). Tal característica inicialmente indicava uma vantagem no uso destas plataformas, pois o suporte era especializado para neste tipo de atividade, bem como os testadores, que constituíam um único ramo de trabalho, diferentemente das demais plataformas que congregam profissionais de diversas áreas. Todas as plataformas foram eliminadas em função do custo. Os valores orçados para execução dos testes nas plataformas foram os seguintes: Utest/Aplause: USD 5.000, Passbrains: USD 250, Testbirds: €1000, crowdtest: R\$ 80 para cada hora de trabalho do analista de testes, acrescido de um valor para cada caso de teste, definido pelo cliente). Os valores levantados foram considerados incompatíveis pela empresa.

Posteriormente passou-se para o levantamento das demais plataformas: Amazon Mechanical Turk e Freelancer. Ambas as plataformas congregam trabalhadores de diversas áreas e não apenas de teste, como nas plataformas anteriores. Em relação ao custo, am-

bas as plataformas se tornaram viáveis, pois permitem que o cliente defina o valor para a execução dos testes. A plataforma Amazon Mechanical Turk foi eliminada, pois o pedido de cadastro foi recusado. Como justificativa a Amazon respondeu que os critérios de revisão de contas são proprietários e que não poderiam informar os motivos para o registro ter sido negado.

Desta forma, selecionou-se para uso a plataforma Freelancer. Como característica da plataforma, a mesma permite a definição de valores de acordo com o tamanho do projeto, o que possibilitou a execução do processo a um menor custo. Além disso, identificou-se na plataforma algumas das características apontadas por Tsai, Wu e Huhns [26]: a plataforma possui painel administrativo para o gerenciamento das tarefas; a plataforma possibilita a comunicação e discussão dos projetos entre os trabalhadores e os clientes; a plataforma possui mecanismos de ranqueamento por meio de reputação, onde são avaliados tanto o trabalhador pela execução do projeto, bem como os clientes, que são avaliados pelo trabalhador contratado; a ferramenta disponibiliza mecanismo de pré-pagamento, em que a remuneração fica retida até a conclusão do projeto pelo trabalhador, o que gera maior segurança para ambas as partes e; a plataforma possui repositório de arquivos, onde foram disponibilizados as instruções e os artefatos necessários para que o trabalhador executasse a tarefa.

Quanto aos atributos de controle de qualidade da plataforma apresentados por Allahbakhsh et al. [24], identificou-se que a plataforma Freelancer possui, quanto à dimensão de perfil do trabalhador, mecanismos de controle da reputação e experiência para os clientes e trabalhadores. Quanto à dimensão relativa ao projeto da tarefa, a plataforma permite a definição da tarefa, bem como a aplicação de restrições, como por exemplo a definição de países específicos para a seleção de trabalhadores para realização da tarefa. Quanto à interface, a plataforma possui painel administrativo, que permite que o cliente gerencie e avalie a submissão da tarefa. Quanto à granularidade, a plataforma não impõe barreiras quanto ao tamanho da tarefa, ficando a definição a cargo do cliente. Por fim, quanto à política de remuneração a plataforma permite a remuneração dos indivíduos por meio de remuneração extrínseca, envolvendo recompensas monetárias e também reputação, o que está diretamente relacionado com o ranqueamento dos trabalhadores em tarefas futuras.

No décimo quarto dia de atividades foi criado um projeto na plataforma Freelancer, incluindo o questionário crowdsourcing como descrição do projeto. Adicionalmente, foram definidas as habilidades necessárias a quem se candidatasse, bem como um valor de orçamento, entre R\$ 30,00 e R\$ 90,00.

Devido ao fato da plataforma Freelancer não dispor de profissionais analistas para gerenciar a execução dos testes, o ator “Analista da plataforma crowdsourcing” não será utilizado.

## Dias 15 e 16

No décimo quinto e décimo sexto dia de atividades foram executadas as atividades “Liberar tarefa de teste” e “Selecionar testador para tarefa de teste”. O projeto foi disponibilizado na plataforma Freelancer para o recebimento de ofertas pelos profissionais. O projeto ficou aberto para ofertas por 24 horas, e neste período foram recebidas 42 ofertas. A maioria destas ofertas foram submetidas por indianos, mas também foram recebidas propostas de bangladechianos, brasileiros, egípcios, japoneses, russos, e americanos. A média de valor das ofertas foi de R\$ 73,00. No estudo piloto considerou-se na seleção da proposta apenas o valor, uma vez que os profissionais com melhor qualificação submeteram propostas com valor superior em comparativo aos sem qualificação na plataforma.

Para a execução da tarefa de teste, selecionou-se a oferta de um profissional indiano, cujo valor proposto foi de R\$ 40,00, com conclusão do projeto em um dia. O profissional não possuía reputação na plataforma, uma vez que não havia realizado nenhum trabalho, e, desta forma, não havia recebido nenhuma avaliação.

## Dia 17

No décimo sétimo dia foi executada a atividade “6. Executar tarefa de teste”, em que o testador executou cada um dos 20 casos de teste associados à tarefa de teste, informando os resultados e gerando registros das falhas identificadas.

## Dias 18 e 19

No décimo oitavo e décimo nono dia de atividades foram executadas as atividades “8. Receber e avaliar resultado da tarefa de teste” e “9. Avaliar falhas reportadas”.

A partir da conclusão e submissão dos resultados da tarefa de teste pelo testador, foram verificados os casos de teste na plataforma Testlink, se todos os casos de teste haviam sido executados e se haviam sido submetidos os registros de falha para os casos de teste que falharam. Dos 20 casos de teste, 10 passaram e 10 falharam. Para o registro das falhas, utilizou-se o modelo de registro de falhas do processo, que foi disponibilizado ao testador. Posteriormente, as falhas identificadas foram avaliadas, registradas e priorizadas pela empresa quanto à necessidade de correção.

O pagamento foi liberado ao testador, bem como foi realizada a avaliação do trabalho na plataforma. Da mesma forma, o contratante também é avaliado na plataforma, ou seja, recebeu-se também uma avaliação do testador quanto à tarefa e o contratante.

Dias 20 a 26

Entre os dias 20 e 26 foram realizadas modificações no processo de teste proposto. A partir da condução do estudo piloto, identificou-se atividades e tarefas que descaracterizavam o caráter adaptável do processo, uma vez que faziam menção a rotinas realizadas, por exemplo, em apenas algumas plataformas crowdsourcing, o que inviabilizaria o uso de outras. Devido a este fato, foram removidas do processo as atividades “Liberar tarefa de teste” e “Selecionar testador para tarefa de teste”. Foi necessária a inclusão da atividade “7. Submeter resultado da tarefa de teste”, dada a necessidade do testador reportar as falhas identificadas, bem como informar a conclusão da execução da tarefa de teste.

Além disso, identificou-se algumas tarefas e atividades cuja execução era opcional, as quais foram retiradas do processo por serem desnecessárias para a condução do mesmo, como por exemplo a priorização dos casos de teste e dos registros das falhas. Por fim, refinou-se a descrição das atividades, tarefas, atores e artefatos do processo, a partir das lacunas identificadas na condução do estudo piloto.

## 5.2 APRESENTAÇÃO DA CONDUÇÃO DO ESTUDO DE CASO EM CADA UNIDADE DE ANÁLISE

A seção a seguir apresenta a condução do estudo de caso em cada uma das unidades de análise. A Subseção 5.2.1 apresenta as fontes de evidência que foram consideradas para a coleta de dados em cada unidade de análise. A Subseção 5.2.2 apresenta a condução do estudo de caso na unidade de análise I e a Subseção 5.2.3 apresenta a condução do estudo de caso na unidade de análise II.

### 5.2.1 Fontes de evidência utilizadas

Foram utilizadas as seguintes fontes de evidência para a coleta de dados do estudo de caso, descritas a seguir:

**Documentação:** São documentos impressos ou eletrônicos existentes nas unidades de análise. Apresentam informações relevantes sobre questões em investigação, como por exemplo descrições de processos e documentações de softwares;

**Registros em arquivo:** Caracterizam-se como documentação em meio digital, podendo estar organizados em diversas formas, como por exemplo arquivos digitais, informações em softwares ou registros e artefatos produzidos na execução do processo;

**Entrevistas:** As entrevistas realizadas durante o estudo de caso basearam-se em conversas guiadas por meio de um questionário previamente realizado. Realizou-se duas

entrevistas em cada unidade de análise, uma no início do estudo de caso, com foco em obter dados para levantamento das organizações e outra ao final, buscando coletar dados e percepções sobre a execução do processo;

**Observação direta:** Visitas de campo realizadas nas unidades de análise, em que apenas observou-se as empresas e não realizaram-se intervenções;

**Observação participante:** Diferencia-se da observação direta pela intervenção do observador no estudo de caso, possibilitando captar a realidade do ponto de vista interno da organização. Algumas atividades do estudo de caso foram realizadas em conjunto, uma vez que em ambas as unidades de análise não haviam profissionais alocados exclusivamente para a realização do estudo de caso.

### 5.2.2 Unidade de análise I

A unidade de análise I, denominada como “empresa A”, é uma empresa de desenvolvimento de software da cidade de Passo Fundo<sup>2</sup>. Possui uma equipe de 20 funcionários. Trabalha com soluções de prateleira, especialmente com softwares de gestão empresarial (ERP), com módulos específicos para empresas do ramo agrícola, micro e pequenas empresas e aplicativos móveis. A empresa utiliza tecnologias comercializadas pela Microsoft Corporation para o desenvolvimento de seus produtos (Windows server, SQL server, ASP.NET, dentre outras).

Em relação ao processo de desenvolvimento de software, a empresa não adota um processo da literatura e desenvolveu um processo próprio. O processo possui como característica um viés tradicional, pois é composto por atividades ordenadas e sequenciais. O processo possui pouca documentação, restrita a um diagrama BPMN, acompanhado de uma breve descrição de cada atividade, sendo que as atividades a serem executadas são conhecidas por todos os colaboradores da empresa de maneira tácita. A empresa desenvolveu um software para a gestão de seus projetos, em que são registradas informações relativas às solicitações de clientes, andamento e histórico das atividades, tempo de execução e estimativas. A empresa não possui padrões de documentação de softwares, e os utiliza de acordo com a necessidade.

Em relação ao teste, a empresa implantou recentemente o uso da plataforma Microsoft Test Manager (MTM)<sup>3</sup> para a gestão de casos de teste. Apesar disso, a empresa

---

<sup>2</sup>Localizada no centro-norte do Estado do Rio Grande do Sul, na região conhecida como Planalto Médio, Passo Fundo destaca-se pela representatividade na área médica, cultural e tecnológica. Com população superior a 183 mil habitantes, é considerada cidade-pólo de mais de 100 municípios localizados na região de abrangência. O conglomerado resulta em uma população de aproximadamente um milhão de pessoas. Referência na área da saúde, Passo Fundo possui um dos mais modernos centros médicos do sul do Brasil. A cidade é considerada emergente na produção de softwares e sua economia está baseada no agronegócio e no setor de prestação de serviços. Fonte: <http://senid.upf.br/index.php/tudo-sobre-o-senid/sobre-passo-fundo>.

<sup>3</sup>Direitos reservados à Microsoft Corporation.

realiza os testes, na maioria das vezes, de maneira exploratória, ou seja, sem o uso de técnicas definidas na literatura. Em relação aos níveis de teste realizados, a empresa realiza testes unitários (realizados pelo programador) e testes funcionais (realizados por um testador). Além disso, a empresa tem interesse em automatizar a execução dos casos de teste, uma vez que a plataforma MTM possui esta funcionalidade.

O software utilizado na execução do processo de teste é um ERP, desenvolvido para controlar os processos operacionais e gerenciais de uma empresa. Possui módulos de cadastros, estoques, compras, produção, comercial, finanças, fiscal/contábil, gerencial, contratos, logística e frotas, insumos e importação e comércio exterior, adquiríveis de acordo com as necessidades específicas dos clientes. Possui ainda, versões do software ERP específicas para empresas do ramo agrícola e para micro empresas. O software é do tipo “aplicação web”, podendo ser instalado em um servidor local na empresa ou em um servidor externo, sendo acessível por meio de navegadores web.

A seguir são descritos os procedimentos de campo planejados no protocolo do estudo de caso, que foram realizados em doze dias, onde foram realizadas visitas à empresa para realização das atividades, alocando períodos de no máximo uma hora e meia de trabalho, a fim de não prejudicar as demais atividades em andamento na empresa.

Adicionalmente, realizou-se na empresa A também o estudo piloto, com o objetivo de refinar os procedimentos a serem adotados no estudo de caso, bem como para verificar se o processo estava adequadamente organizado e documentado.

#### Dia 1

No primeiro dia de atividades na empresa realizou-se uma reunião de apresentação e planejamento das atividades do estudo de caso. Foram apresentadas as mudanças no processo de teste, realizadas a partir das necessidades anteriormente constatadas na condução do estudo piloto. Essas mudanças foram então repassadas à representação do processo da empresa integrado ao processo de teste.

Posteriormente, foi definido o software que seria utilizado para a execução das atividades do processo e foram obtidos os recursos necessários. Optou-se, da mesma forma que no estudo piloto, pela execução de testes em dois módulos do software ERP, de cadastros e comercial.

#### Dias 2, 3 e 4

No segundo, terceiro e quarto dia de atividades na empresa foram executadas as atividades do processo: “1. Elaborar plano de teste crowdsourcing” e “2. Definir os cenários de teste”.

Utilizou-se o modelo de plano de teste disponibilizado pelo processo, uma vez que a empresa não possui documento padronizado próprio para adaptação. A documentação dos requisitos, artefato de entrada desta atividade, teve que ser elaborada, uma vez que a empresa não adota o uso de documentação de requisitos. Utilizou-se a abordagem de casos de uso para representar os requisitos do software. Com base nos casos de uso, foi possível definir os cenários de teste. Foram elencados os seguintes cenários: Dentro do módulo de cadastros, selecionou-se o cenário “Cadastro de cliente / fornecedor (informações complementares)”, e dentro do módulo comercial optou-se pelos cenários “cadastro de pedidos” e “inclusão de pré-venda”.

#### Dias 5 e 6

No quinto e sexto dia de atividades na empresa foi executada a atividade do processo: “3. Elaborar os casos de teste”. Para a elaboração dos casos de teste, inicialmente utilizou-se o modelo de caso de teste disponibilizado no processo. Para o estudo de caso foram elaborados 30 casos de testes para os cenários de teste elencados, utilizando os critérios de particionamento em classes de equivalência, análise do valor limite, teste funcional sistemático e error guessing. Posteriormente, os casos de teste foram traduzidos para o idioma inglês e inseridos na plataforma testlink, utilizada para o registro da execução dos casos de teste. O uso da plataforma testlink foi necessário, uma vez que para o uso do Microsoft Test Manager (plataforma que a empresa A utiliza para gestão dos casos de teste) é necessário instalá-lo no computador do testador, bem como configurar o acesso ao servidor da empresa, o que não foi possível, uma vez que o acesso só é possível na rede local da empresa.

#### Dia 7

No sétimo dia de atividades na empresa foi realizada a atividade “4. Configurar ambiente de teste”.

Em relação à configuração do ambiente de teste, utilizou-se a estrutura organizada para o estudo piloto (software instalado em um servidor exclusivamente para a execução de testes, usuário fictício no software e permissões de acesso limitadas apenas ao necessário para a execução dos casos de teste). Devido ao software-alvo do processo ser uma aplicação web, o mesmo foi disponibilizado ao testador por meio do endereço (URL), informado em cada caso de teste.

## Dia 8

No oitavo dia de atividades na empresa foi realizada a atividade “5. Especificar tarefa de teste”. A especificação da tarefa de teste foi realizada a partir das informações do questionário crowdsourcing, disponíveis no plano de teste. Da mesma forma que no estudo piloto, utilizou-se a plataforma Freelancer, onde foi criado um projeto que incluiu as informações presentes no questionário crowdsourcing como descrição do projeto. Adicionalmente, foram definidas as habilidades necessárias a quem se candidatasse ao projeto, bem como um valor de orçamento, entre R\$ 30,00 e R\$ 90,00.

## Dia 9 e 10

No nono e décimo dia de atividades foram executadas as atividades do processo “6. Executar tarefa de teste” e “7. Submeter resultado da tarefa de teste”.

No nono dia o projeto foi disponibilizado na plataforma Freelancer para o recebimento de ofertas pelos profissionais. O projeto ficou aberto para ofertas por 12 horas, e neste período foram recebidas 17 ofertas. A maioria destas ofertas novamente foram realizadas por indianos, mas também foram recebidas propostas de canadenses, alemães, turcos, ucranianos e americanos. A média de valor das ofertas foi de R\$ 67,00. Para a seleção da proposta nesta unidade de análise considerou-se apenas o valor, uma vez que os profissionais com melhor qualificação submeteram propostas com valor superior em comparativo aos sem qualificação na plataforma.

Para a execução da tarefa de teste, selecionou-se a oferta de um profissional indiano, cujo valor proposto foi de R\$ 40,00, com conclusão do projeto em um dia. O profissional não possuía reputação na plataforma, uma vez que não havia realizado nenhum trabalho, e, desta forma, não havia recebido nenhuma avaliação.

## Dia 11

No décimo primeiro dia da condução do estudo de caso foram executadas as atividades do processo “8. Avaliar execução da tarefa de teste” e “9. Avaliar falhas reportadas”.

A partir da conclusão e submissão dos resultados da tarefa de teste pelo testador, foram verificados os casos de teste na plataforma Testlink, se todos os casos de teste haviam sido executados e se haviam sido submetidos os registros de falha para os casos de teste que falharam. Dos 30 casos de teste, 17 passaram e 13 falharam. Para o registro das falhas, utilizou-se o modelo de registro de falhas do processo, que foi disponibilizado previamente ao testador. Posteriormente, as falhas identificadas foram avaliadas e registradas pela empresa quanto à necessidade de correção.

O pagamento foi liberado ao testador, bem como foi realizada a avaliação do trabalho na plataforma. Da mesma forma, o contratante também foi avaliado na plataforma, ou seja, recebeu-se também avaliação do testador quanto à tarefa e ao contratante.

Dia 12

No último dia de atividades realizou-se uma reunião de conclusão dos trabalhos, guiada por um roteiro de avaliação semi-estruturado (Apêndice D). Neste roteiro de avaliação foram consideradas várias dimensões do trabalho realizado, buscando respostas para as questões elencadas no protocolo e proposições definidas no projeto do estudo de caso.

### **5.2.3 Unidade de análise II**

A unidade de análise II, denominada como “empresa B”, é uma empresa de desenvolvimento de software da cidade de Passo Fundo. Possui uma equipe de 8 funcionários. Trabalha com soluções sob demanda (necessidades específicas de clientes) e também com soluções de prateleira, como o software de controle de imobiliárias e website e-commerce de comercialização de fotos. Desta forma, não possui uma estrutura definida de plataformas de desenvolvimento de software (IDEs, SGBDs, Linguagem de programação), utilizando tecnologias de acordo com as necessidades específicas de cada projeto.

Em relação ao processo de desenvolvimento de software, a empresa não adota um processo da literatura e desenvolveu um processo próprio. O processo possui como característica um viés tradicional, pois é composto por atividades ordenadas e sequenciais. O processo não possui documentação, sendo que as atividades a serem executadas são conhecidas por todos os colaboradores da empresa de maneira tácita. A empresa desenvolveu um software para a gestão de seus projetos, em que são registradas informações relativas às solicitações de clientes, andamento e histórico das atividades, tempo de execução e estimativas. A empresa não possui padrões de documentação de softwares, e os utiliza de acordo com a necessidade.

Em relação ao teste dos softwares desenvolvidos, a empresa os realiza, na maioria das vezes, de maneira exploratória, ou seja, sem o uso de técnicas definidas na literatura e sem o apoio de softwares de gestão de casos de teste e de falhas. Em relação aos níveis de teste realizados, a empresa utiliza testes unitários (realizados pelo programador) e testes funcionais (realizados por um testador). Apesar disso, a empresa está em processo de ampliar o uso de testes, uma vez que estão em estudo o uso de softwares de gestão de casos de teste e de registro de falhas.

Especificamente em relação ao teste funcional, a empresa atualmente utiliza um documento de texto para estabelecer um breve planejamento dos testes, em que são elen-

cados os cenários de teste e os testes que deverão ser realizados, sem descrevê-los por meio de casos de teste. O registro de falhas é realizado diretamente no software de gestão da empresa.

O software utilizado na execução do processo de teste é um software de comércio de fotografias digitais. O software permite a personalização do site pelo fotógrafo, permitindo a gestão e controle dos eventos com relatórios gerenciais dos acessos, visualizações e comercializações. O software permite que o fotógrafo disponibilize as fotos para escolha pelos clientes, bem como efetue a venda das fotos pela internet, podendo estas serem entregues em formato digital ou impresso, neste caso, enviadas via correio. Para o cliente, o software permite comodidade e facilidade na escolha das fotos, podendo adquiri-las em múltiplos tamanhos e formatos: digital e impresso. Além disso, o software permite também o pagamento por meio do site ou diretamente ao fotógrafo.

A seguir é descrito o relato do procedimento de campo realizado na empresa B. Os procedimentos de campo planejados no protocolo do estudo de caso foram realizados em 12 dias, em que ocorreram visitas à empresa para realização das atividades, alocando períodos de no máximo uma hora e meia de trabalho, a fim de não prejudicar as demais atividades em andamento na empresa.

#### Dia 1

O procedimento de campo na empresa B foi iniciado com a reunião de apresentação e planejamento das atividades. Inicialmente foi apresentado o objetivo do trabalho que seria realizado na empresa, bem como o planejamento por meio dos procedimentos de campo que seriam realizados durante o período.

Na sequência realizou-se o levantamento de informações da empresa, por meio de uma entrevista guiada por um roteiro semi-estruturado (Apêndice C), composto por perguntas definidas anteriormente e por questões que surgissem no decorrer da mesma. Com base nesta entrevista foi possível estabelecer um panorama geral da empresa, suas características e particularidades e estabelecer a representação do processo de desenvolvimento de software que a empresa utiliza, por meio de um diagrama BPMN.

#### Dia 2

No segundo dia de atividades na empresa foi apresentado o processo de teste, suas atividades, tarefas, atores e artefatos, utilizando a documentação desenvolvida para suporte ao processo. Posteriormente, realizou-se a integração do processo de teste com o processo em uso na empresa (levantado no dia anterior), com auxílio do “Roteiro para implantação do processo”. Por fim, elaborou-se um diagrama BPMN representando a im-

plantação do processo de teste proposto juntamente com o processo que a empresa utiliza. Este diagrama serviu posteriormente como base para a execução do processo.

Por fim, foi definido o software que seria utilizado para a execução das atividades do processo e foram obtidos os recursos necessários. Optou-se por um software e-commerce em desenvolvimento na empresa, que permite a comercialização de fotografias. O software oferece ao cliente (fotógrafo): venda de suas fotos pela internet, criação de site personalizado e integrado à redes sociais, canal para comunicação e atendimento aos clientes e painel de gestão e controle dos eventos.

Dias 3 e 4

No terceiro e quarto dia de atividades na empresa foram executadas as atividades do processo: “1. Elaborar plano de teste crowdsourcing” e “2. Definir os cenários de teste”.

Em relação à primeira atividade, utilizou-se o modelo de plano de teste disponibilizado pelo processo, uma vez que a empresa não possui documento padronizado próprio para adaptação. A documentação dos requisitos, artefato de entrada desta atividade, foi disponibilizada pela empresa, que utiliza a abordagem de casos de uso para representar os requisitos do software. Com base nos casos de uso, foi possível definir os cenários de teste. Foram elencados os seguintes cenários de teste: Cadastro de papel, cadastro de tamanho de papel, cadastro de pacotes promocionais, configuração do site do fotógrafo, cadastro de cliente e pedido e acompanhamento do pedido de fotografias.

Dias 5, 6 e 7

No quinto, sexto e sétimo dia de atividades na empresa foi executada a atividade do processo: “3.Elaborar os casos de teste”.

Para a elaboração dos casos de teste, inicialmente utilizou-se o modelo de caso de teste disponibilizado no processo. Foram elaborados 30 casos de testes para os cenários de teste elencados anteriormente, com número de passos entre 3 e 23, utilizando os critérios de particionamento em classes de equivalência, análise do valor limite, teste funcional sistemático e error guessing. Posteriormente, os casos de teste foram traduzidos para o idioma inglês e inseridos na plataforma Testlink, utilizada para o registro da execução dos casos de teste.

Dia 8

No oitavo dia de atividades na empresa foram executadas as atividades do processo: “4. Configurar ambiente de teste” e “5. Especificar tarefa de teste”.

Em relação à configuração do ambiente de teste, foi necessário criar um cadastro para teste no software, bem como contas de e-mail fictícias para testar notificações de andamento de pedidos de fotografias, que são enviadas automaticamente para o e-mail do cliente e do fotógrafo, a cada alteração em um pedido. Devido ao software-alvo do processo ser uma aplicação web, o mesmo foi disponibilizado em um servidor da empresa e o endereço (URL) de acesso ao software foi repassado em cada caso de teste.

A especificação da tarefa de teste foi realizada a partir das informações do questionário crowdsourcing, disponíveis no plano de teste. Utilizou-se novamente a plataforma Freelancer para o recrutamento do testador. Foi criado um novo projeto na plataforma, incluindo o questionário crowdsourcing como descrição do projeto. Adicionalmente, foram definidas as habilidades necessárias a quem se candidatasse ao projeto, bem como um valor de orçamento, entre R\$ 45,00 e R\$ 90,00 (ou R\$ 1,50 a R\$ 2,50 para cada caso de teste).

Dias 9 e 10

No nono e décimo dia de atividades foram executadas as atividades do processo “6. Executar tarefa de teste” e “7. Submeter resultado da tarefa de teste”.

No nono dia o projeto foi disponibilizado na plataforma Freelancer para o recebimento de ofertas pelos profissionais. O projeto ficou aberto para ofertas por pouco mais de 24 horas e, neste período foram recebidas 72 ofertas. A maioria destas ofertas foram realizadas por indianos, mas também foram recebidas propostas de paquistaneses, russos, romenos, irlandeses, americanos, venezuelanos, canadenses, ucranianos, vietnamitas, egípcios, búlgaros, britânicos, bielorrussos e poloneses. A média de valor das ofertas foi de R\$ 76,00. Diferentemente da unidade de análise anterior, para a seleção da proposta foi considerado, além do preço, a reputação do testador, uma vez que houveram mais propostas, bem como profissionais melhor qualificados do que anteriormente e a um custo menor.

Para a execução da tarefa de teste, selecionou-se a oferta de um profissional indiano, cujo valor proposto foi de R\$ 50,00, com conclusão do projeto em um dia. Sua reputação na plataforma Freelancer é de 4.9 (em uma escala que vai de 0 a 5), com base em 21 avaliações de outros contratantes.

Dia 11

No décimo primeiro dia foram executadas as atividades do processo “8. Avaliar execução da tarefa de teste” e “9. Avaliar falhas reportadas”.

A partir da conclusão e submissão dos resultados da tarefa de teste pelo testador, foram verificados os casos de teste na plataforma Testlink, se todos os casos de teste haviam sido executados e se haviam sido submetidos os registros de falha para os casos de teste que falharam. Dos 30 casos de teste, 21 passaram e 9 falharam. Para o registro das falhas, utilizou-se o modelo de registro de falhas do processo, que foi disponibilizado ao testador. Posteriormente, as falhas identificadas foram avaliadas e registradas pela empresa quanto à necessidade de correção.

O pagamento foi liberado ao testador, bem como foi realizada a avaliação do trabalho na plataforma. Da mesma forma, o contratante também é avaliado na plataforma, ou seja, recebeu-se também avaliação do testador quanto à tarefa e ao contratante.

Dia 12

No último dia de atividades na empresa realizou-se uma reunião de conclusão dos trabalhos, guiada por um roteiro de avaliação semi-estruturado (Apêndice D). Neste roteiro de avaliação foram consideradas várias dimensões do trabalho realizado, buscando respostas para as questões elencadas no protocolo e proposições definidas no projeto do estudo de caso.

### 5.3 ANÁLISE E DISCUSSÃO DAS PROPOSIÇÕES

Esta seção apresenta a análise e discussão das proposições levantadas no projeto de pesquisa do estudo de caso. Cada proposição foi avaliada a partir das informações obtidas com a coleta de dados durante a condução do estudo de caso nas unidades de análise, juntamente com as questões definidas no protocolo, que apresentam as respectivas fontes de evidência que foram utilizadas (documentação, registros em arquivo, entrevista, observação direta e observação participante). A seguir cada proposição do estudo de caso é apresentada, bem como as evidências coletadas que confirmam, refutam ou mantêm inconclusivas suas afirmações.

**Proposição 1:** As empresas são capazes de realizar testes funcionais utilizando o processo proposto.

Situação: **Confirmada.** A partir da observação direta percebeu-se que a documentação do processo disponibilizada às unidades de análise foi suficiente para que estas conseguissem realizar testes funcionais utilizando o processo. Adicionalmente, os registros em arquivo da execução do processo corroboram com a afirmativa, uma vez que em cada unidade de análise elaborou-se 30 casos de testes, que foram posteriormente executados. Apesar disso, algumas tarefas do processo foram realizadas por meio da observação participante, em que o observador intervém e participa da execução do processo, o que de certa

forma causa interferência na execução das atividades, podendo ter facilitado sua execução em ambas as unidades de análise.

**Proposição 2:** O processo é genérico, ou seja, as empresas são capazes de integrar o processo de teste ao processo de desenvolvimento já em execução.

Situação: **Confirmada.** A partir da observação direta percebeu-se que a documentação do processo disponibilizada às unidades de análise foi suficiente para que estas conseguissem integrá-las ao processo de desenvolvimento de software que utilizam. Adicionalmente, os registros em arquivo da execução do processo corroboram com a afirmativa, uma vez que em cada unidade de análise foram elaborados diagramas BPMN do processo da empresa integrado ao processo proposto.

**Proposição 3:** Os testadores geograficamente dispersos são capazes de executar as tarefas de teste.

Situação: **Confirmada.** A partir da observação participante percebeu-se que os testadores conseguiram executar corretamente as tarefas de teste, seguindo os passos solicitados nos casos de teste e gerando registros em arquivo, como resultados de teste (passou ou falhou) e registros de falhas, por exemplo.

**Proposição 4:** Crowdsourcing é uma abordagem viável para o teste funcional.

Situação: **Confirmada.** No decorrer da condução do estudo de caso percebeu-se que a execução do processo de teste ocorreu de maneira satisfatória, em ambas as unidades de análise. O uso da técnica de observação participante permitiu o acompanhamento minucioso da execução do processo, onde foi possível identificar que as atividades foram executadas conforme descritas na documentação do processo. Foi de grande relevância a condução do estudo piloto, anteriormente à execução do estudo de caso, pois percebeu-se que as mudanças aplicadas corrigiram as inconsistências anteriormente identificadas.

Também é importante destacar que não houveram problemas em relação aos testadores, que executaram corretamente os casos de teste e reportaram as falhas identificadas, bem como no uso da plataforma e das ferramentas e artefatos, que permitiram e deram suporte à execução do processo. Além disso, os registros em arquivo obtidos com a condução do estudo de caso corroboram com a afirmativa, uma vez que foram executados 30 casos de teste em cada unidade de análise.

Por fim, esta proposição deve ser avaliada em conjunto com as demais, uma vez que também apresentam evidências quanto à viabilidade do uso do crowdsourcing para o teste funcional.

**Proposição 5:** Crowdsourcing é uma abordagem barata de executar testes funcionais.

Situação: **Inconclusiva.** O custo é relativo, pois existem vários fatores envolvidos como por exemplo: o software em teste, a demanda da empresa, a quantidade de casos de teste e a frequência em que são executados. Apesar disso, durante a entrevista de

conclusão, em ambas as unidades de análise, os entrevistados afirmaram considerar a proposta mais barata, na situação em que há uma baixa demanda de testes que justifique a contratação de um novo profissional, uma vez que pequenas empresas de desenvolvimento de software possuem demanda variável, especialmente em relação a testes.

**Proposição 6:** O resultado recebido ao final da tarefa de teste é relevante.

Situação: **Confirmada.** A partir da observação participante identificou-se que, tanto no estudo piloto quanto nas unidades de análise, o resultado recebido para cada tarefa de teste foi adequado, uma vez que todos os casos de teste foram executados, gerando registros de falha que foram reportados para todos os casos de teste que indicaram resultado como “falha”. Em entrevista realizada ao final do estudo de caso, na unidade de análise II considerou-se adequado, enquanto na unidade de análise I considerou-se adequado com ressalvas, uma vez que em alguns casos de teste os testadores identificaram falhas “falso positivo”, pois um software do tipo ERP possui diversas parametrizações, as quais estavam incorretamente configuradas na versão utilizada durante o estudo de caso. Por fim, na unidade de análise I concluiu-se que é necessário um cuidado maior na especificação do caso de teste e na parametrização do software para que os resultados dos testes sejam fidedignos.

**Proposição 7:** O idioma não é barreira para o uso do processo nas empresas.

Situação: **Refutada.** Durante a observação participante identificou-se que em ambas as unidades de análise os colaboradores das empresas não possuem conhecimento adequado do idioma inglês, o que prejudica a escrita dos casos de teste, bem como a negociação com os testadores. Como as plataformas tem alcance mundial, todo o processo de negociação com os profissionais e a execução das tarefas é realizada por meio do idioma inglês. Adicionalmente, durante entrevista realizada ao final do estudo de caso, em ambas as unidades de análise concluiu-se que o uso do inglês como idioma é uma barreira, pois dificulta a execução das tarefas nas empresas, a negociação com testadores, bem como aumenta o tempo demandado, especialmente na escrita dos casos de teste.

**Proposição 8:** O uso das plataformas crowdsourcing não é barreira para o uso do processo nas empresas.

Situação: **Inconclusiva.** Durante a observação participante identificou-se em ambas as unidades de análise que não houve problemas no uso da plataforma Freelancer. Porém, para responder esta pergunta seria necessário realizar um estudo de caso com cada uma das plataformas existentes, verificando quais são adequadas ou não. Adicionalmente, na unidade de análise I, durante entrevista realizada ao final do estudo de caso, concluiu-se que o problema maior está em definir a plataforma crowdsourcing mais adequada ao contexto, uma vez que as empresas não possuem critérios para avaliar, bem como não saberiam como proceder a escolha da plataforma, uma vez que o processo não dá suporte para esta decisão.

**Proposição 9:** A proteção da propriedade intelectual não é barreira pra o uso do processo nas empresas.

Situação: **Inconclusiva.** Durante a execução do processo em ambas as unidades de análise utilizou-se para teste softwares ou módulos de software que não apresentavam riscos em relação à propriedade intelectual, bem como utilizaram-se dados fictícios para os testes. Imaginou-se, durante a elaboração das proposições, que a propriedade intelectual não seria um problema, pelo fato do teste funcional ser uma técnica “caixa preta”. Porém, durante entrevista realizada ao final do estudo de caso na unidade de análise I, concluiu-se que a segurança da propriedade intelectual é uma barreira para o uso do processo nas empresas, pois não há garantias que os testadores não irão tentar ter acesso a dados não autorizados ou tentar burlar os mecanismos de segurança dos softwares em teste.

**Proposição 10:** O uso de reputação e preço são relevantes na escolha do testador.

Situação: **Confirmada.** Durante a observação participante identificou-se que em ambas as unidades de análise os dois critérios foram levados em consideração. Apesar disso, no estudo piloto e na unidade de análise I optou-se pela seleção de um testador sem reputação na plataforma, em função de que os profissionais com boa reputação haviam feito propostas com valor muito superior em comparativo aos não qualificados. Na unidade de análise II selecionou-se a proposta de um testador que aliou uma boa reputação e preço adequado. Apesar de serem utilizados testadores sem reputação para execução dos testes, não foram percebidas diferenças em relação ao trabalho realizado pelos profissionais, uma vez que novos trabalhadores na plataforma só passam a ter reputação quando são contratados, ou seja, todos os profissionais iniciam sem reputação e vão sendo avaliados a cada tarefa que realizam nas plataformas.

**Proposição 11:** O meio de pagamento e a segurança da transação não são barreiras para o uso do processo nas empresas.

Situação: **Inconclusiva.** Durante observação participante, em ambas as unidades de análise, identificou-se certa insegurança em relação ao pagamento. Esta insegurança foi superada durante o uso da plataforma Freelancer, uma vez que a mesma possui mecanismo de pré-pagamento, em que a remuneração fica retida pela plataforma até que o testador conclua a atividade e o cliente libere o pagamento, caso considere que a tarefa foi executada corretamente. Adicionalmente, durante entrevista realizada ao final da execução do processo, em ambas as unidades de análise, considerou-se que o mecanismo de pagamento e a segurança da transação é adequada. Por fim, a resposta para esta questão é inconclusiva, pois o meio de pagamento e a segurança da transação variam de acordo com a plataforma crowdsourcing selecionada.

**Proposição 12:** O processo é suficientemente documentado para o entendimento por parte das empresas.

Situação: **Confirmada.** Durante observação direta e observação participante em ambas as unidades de análise percebeu-se que não houveram grandes problemas no entendimento da documentação do processo. Dúvidas pontuais foram esclarecidas por meio do acesso ao material disponibilizado no site. Além disso, durante o estudo de caso piloto identificou-se inconsistências na documentação do processo, que foram posteriormente corrigidas. Como exemplo de inconsistência identificada pode-se citar seções do plano de teste desnecessárias e que burocratizavam o processo. Por fim, nas entrevistas realizadas ao final dos procedimentos de campo em cada unidade de análise os participantes consideraram a documentação para apoio do processo adequada, simplificada e compreensível; consideraram a descrição das atividades e tarefas do processo adequada e sucinta e; consideraram que os diagramas disponibilizados juntamente com as atividades estavam organizados, seguindo uma ordem lógica e orientada, porém, por si só não explicavam o processo, o que indica a importância do diagrama estar acompanhado da descrição das atividades e tarefas para o entendimento do processo.

**Proposição 13:** O processo permite o uso de qualquer técnica de teste funcional (análise de valor limite, partição de equivalência, dentre outras).

Situação: **Confirmada.** Durante a execução do processo, em ambas as unidades de análise, utilizou-se para a escrita dos casos de teste as técnicas de partição de equivalência, análise de valor limite, teste funcional sistemático e error guessing, que se adequavam aos cenários de teste definidos. Por meio da observação participante, não identificou-se barreiras no processo para o uso das técnicas de teste funcional, uma vez que a construção do processo não restringe ao testador o uso das técnicas. A única restrição existente está no analista de testes, que necessita conhecer as técnicas de teste funcional para utilizá-las adequadamente.

**Proposição 14:** O processo é executado em um tempo adequado.

Situação: **Inconclusiva.** Durante a execução do processo, em ambas as unidades de análise, as atividades foram executadas de acordo com a disponibilidade da empresa, sendo que na maioria das vezes dedicou-se uma hora diária durante os doze dias dedicados, em cada empresa, para a execução das atividades. A partir da observação direta percebeu-se que a execução do processo depende do entendimento do mesmo. Como o estudo de caso piloto foi executado na mesma empresa que a unidade de análise I, percebeu-se que houve um ganho de tempo em relação à primeira execução do processo. Por fim, durante entrevista realizada ao final dos procedimentos de campo em cada unidade de análise os participantes consideraram que o tempo de execução das atividades é adequado em cerca de 80% dos casos, uma vez que existem situações em que há urgência de entrega do software, e neste caso a execução dos testes na empresa pode ser mais rápida em comparativo com a abordagem do processo.

**Proposição 15:** O processo flexibiliza a demanda de mão de obra de profissionais de teste, de acordo com a demanda da empresa.

Situação: **Confirmada.** Durante a condução do estudo de caso, em ambas as unidades de análise, percebeu-se que a demanda de testes é variável em pequenas empresas que desenvolvem software. Desta forma, o processo auxilia na flexibilização da mão de obra em relação à demanda de trabalho, uma vez que não há vínculo empregatício com os testadores e sua contratação é ocasional. Por fim, durante entrevista realizada ao final dos procedimentos de campo em cada unidade de análise, os participantes consideraram que a flexibilização da mão de obra é um aspecto importante e um atrativo ao uso do processo por pequenas empresas de desenvolvimento de software, que possuem pouca demanda que justifique a contratação de um testador.

#### 5.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Este capítulo apresentou a avaliação do processo de teste proposto neste trabalho a partir da condução de um estudo de caso, composto por um contexto e duas unidades de análise.

Na elaboração do projeto de pesquisa do estudo de caso foram levantadas quinze proposições, sendo que todas foram contempladas e respondidas durante a avaliação do estudo de caso. A Tabela 1 compila os resultados obtidos.

Tabela 1. Situação das proposições após o estudo de caso.

<b>Proposição</b>	<b>Resposta</b>
Proposição 1	Confirmada
Proposição 2	Confirmada
Proposição 3	Confirmada
Proposição 4	Confirmada
Proposição 5	Inconclusiva
Proposição 6	Confirmada
Proposição 7	Refutada
Proposição 8	Inconclusiva
Proposição 9	Inconclusiva
Proposição 10	Confirmada
Proposição 11	Inconclusiva
Proposição 12	Confirmada
Proposição 13	Confirmada
Proposição 14	Inconclusiva
Proposição 15	Confirmada

Ao final da condução do estudo de caso identificou-se que 9 proposições levantadas foram confirmadas, ou seja, confirmaram-se as hipóteses apresentadas.

Uma proposição foi refutada, relativa ao uso do idioma inglês para a execução do processo nas empresas. Imaginou-se que isto não seria uma barreira, porém o estudo de caso evidenciou que este é um elemento que dificultará a adoção do processo em pequenas empresas de desenvolvimento de software.

Por fim, cinco proposições, ao final da condução do estudo de caso, permaneceram inconclusivas. As hipóteses não foram nem confirmadas nem refutadas, uma vez que as informações levantadas no estudo de caso não foram suficientes para a avaliação conclusiva. Tais proposições poderão ser novamente avaliadas em estudos futuros, para que recebam respostas conclusivas a partir de mais elementos.

## 6. CONSIDERAÇÕES FINAIS

Este trabalho teve como principal objetivo o desenvolvimento de um processo para a execução de testes funcionais utilizando crowdsourcing. Os objetivos específicos inicialmente delineados foram alcançados no decorrer da execução das atividades previstas.

Os três primeiros objetivos específicos envolviam o estudo de teste de software, bem como do uso de crowdsourcing no desenvolvimento e no teste de software. A etapa de revisão teve um importante papel na realização deste trabalho, uma vez que o crowdsourcing é um tema emergente, o que exigiu uma ampla pesquisa dos trabalhos já realizados nesta área, para verificar iniciativas existentes. Nesta pesquisa não foram identificados trabalhos que relatassem a existência de processos que utilizassem crowdsourcing para a execução de testes funcionais, porém foram identificadas iniciativas que utilizam esta abordagem para outras categorias de teste, o que motivou esta pesquisa.

Para a realização do terceiro objetivo específico, que envolvia propor um processo de teste funcional baseado em crowdsourcing, foi necessária a realização das etapas de investigação e ação. A imersão em uma empresa de desenvolvimento de software, realizada na etapa de investigação, teve um importante papel em identificar as dificuldades para implantar um processo de teste em uma empresa. Nesta imersão percebeu-se que as empresas possuem dificuldades em alocar mão de obra para testes, em função dos custos para manter um testador, o qual possui, muitas vezes, demanda de trabalho variável.

O desenvolvimento do processo de teste, principal contribuição deste trabalho, foi um grande desafio. Buscou-se, durante elaboração do processo na etapa de ação, identificar os elementos mínimos necessários, propondo um processo adaptável quanto a documentação, ferramentas e plataformas utilizadas, permitindo que seja possível a sua integração aos processos de desenvolvimento de software já em uso pelas empresas.

A opção pelo teste funcional mostrou-se adequada, devido ao fato de ser uma abordagem de teste sob o ponto de vista do usuário, o que facilita o entendimento e execução pelo testador. Adicionalmente, a descrição do teste por meio de casos de teste permite a execução pelo testador, mesmo este não possuindo conhecimentos específicos sobre a aplicação em teste.

O uso da notação BPMN foi relevante na documentação do processo, uma vez que desta forma foi possível padronizar a representação dos elementos, facilitando o entendimento das fases, atividades, tarefas, atores e artefatos por todos os envolvidos na execução do processo.

Para alcançar o quinto objetivo específico proposto, que envolvia a avaliação do processo, foi necessária a condução de um estudo de caso, realizada na etapa de avaliação. A realização do estudo de caso demonstrou que o processo de teste proposto pode ser executado em pequenas empresas de desenvolvimento de software.

Como estratégia para análise dos resultados, foram elaboradas quinze proposições, as quais foram avaliadas a partir das evidências coletadas nas duas unidades de análise, que representam pequenas empresas de desenvolvimento de software da cidade de Passo Fundo. Destas proposições, nove foram confirmadas, cinco permaneceram inconclusivas e uma foi refutada. Especialmente em relação as proposições que permanecem inconclusivas, são necessários novos estudos, para que as proposições sejam efetivamente confirmadas ou refutadas, uma vez que as evidências levantadas na condução do estudo de caso não foram suficientes para tal.

Durante a condução do estudo de caso percebeu-se a presença das características do uso de crowdsourcing para teste, descritas na revisão bibliográfica. Diversos testadores foram mobilizados para a realização dos testes, com vários perfis distintos. Os testadores selecionados apresentaram propostas para a execução das tarefas, que foram cumpridas. Houve flexibilização da mão de obra, bem como o pagamento pela execução das tarefas de testes ocorreu pela produtividade dos testadores. A execução dos testes ocorreu em um ambiente externo à empresa, simulando um ambiente real de utilização do software em teste. Foram elaborados modelos de documentação, que padronizaram os resultados obtidos com a execução das tarefas de teste. Percebeu-se também que as nove atividades propostas no processo foram suficientes para sua execução. Adicionalmente, as falhas identificadas foram validada, e foi avaliada a execução da tarefa de teste pelo cliente, utilizando as ferramentas de reputação da plataforma.

Apesar de não ter sido utilizada no estudo de caso uma plataforma crowdsourcing específica para teste, o uso da plataforma Freelancer foi válido. As várias ofertas de testadores para as tarefas de teste demonstraram a existência de uma comunidade de testadores na plataforma. Os mecanismos de reputação e de pagamento em espera proporcionam segurança na execução da tarefa de teste; o testador só recebeu a remuneração após a execução da tarefa ser aprovada pelo cliente, bem como a reputação registrada no perfil do profissional foi observada pelo cliente na seleção do testador para a execução da tarefa.

A opção pelo estudo de caso para avaliação do processo foi válida, pois permitiu a execução do processo em duas empresas de desenvolvimento de software da cidade de Passo Fundo, simulando o uso do processo em um ambiente real. Da mesma forma, o uso de softwares em desenvolvimento pelas empresas permitiu a realização de testes reais, proporcionando às empresas participantes a identificação e correção de falhas nos softwares que comercializam.

## 6.1 DIFICULDADES ENCONTRADAS

A falta de bibliografia foi uma das dificuldades encontradas na realização deste trabalho. A principal obra sobre crowdsourcing aborda seu uso nas mais diversas áreas do conhecimento, como por exemplo a tradução de textos, a elaboração colaborativa de estampas de camisetas e o desenvolvimento de novas ideias inovadoras para empresas que desejam melhorar seus produtos. Especificamente sobre o uso de crowdsourcing na engenharia de software, o livro aborda superficialmente o seu surgimento, ligado ao desenvolvimento colaborativo de softwares de código aberto, bem como a existência de algumas plataformas crowdsourcing.

Em relação aos trabalhos realizados que envolvem testes funcionais sendo executados por meio da abordagem crowdsourcing, percebeu-se a ausência de referências que discorram sobre o seu uso, atrelado a um processo de teste. As poucas iniciativas existentes nesta área abordam pontos específicos do teste, ou então apresentam processos simplificados e superficiais.

Outra dificuldade encontrada envolveu o levantamento das plataformas crowdsourcing. Buscou-se o contato com diversas plataformas, a fim de levantar informações sobre seu uso e de como poderiam ser integradas a um processo de teste, porém de algumas, não houve retorno. Outro problema identificado foram os altos custos do uso de algumas plataformas, levando-se em consideração o uso por pequenas empresas de desenvolvimento de software. Também é importante salientar os bloqueios impostos por algumas plataformas, como por exemplo a Amazon Mechanical Turk, que recusou o cadastro, bem como não revelou a razão pela qual o pedido de cadastro foi rejeitado.

## 6.2 TRABALHOS FUTUROS

Ao término do presente trabalho, identificou-se algumas questões que podem ser pesquisadas e desenvolvidas em trabalhos futuros.

A primeira delas envolve a melhoria do ambiente de documentação do processo disponibilizado, envolvendo a criação de um portal colaborativo, que permita a colaboração das empresas e trabalhadores na melhoria do processo e de sua documentação. Também é necessária a adoção de alguma estratégia para a gestão e monitoramento do acesso e uso do processo, a fim de identificar empresas e pessoas interessadas em utilizá-lo, bem como manter o controle por meio de um portfólio de empresas que estiverem utilizando o mesmo.

A condução do estudo de caso apontou cinco proposições cuja afirmação ainda é inconclusiva. Trabalhos futuros podem avaliar novamente tais proposições, buscando novas informações para efetivamente confirmá-las ou refutá-las.

Além disso, também é necessário o estudo da aplicação do processo com outras plataformas crowdsourcing, uma vez que apenas a plataforma Freelancer foi utilizada na condução do estudo de caso realizado neste trabalho.

Outro passo importante para a sequência deste trabalho é a aplicação de alguma técnica com foco na melhoria do processo, como por exemplo o PDCA (Plan–Do–Check–Act) ou o GQM (Goal Question Metric).

Ainda, pesquisas futuras também podem avaliar os aspectos legais do uso do crowdsourcing em relação à legislação, uma vez que a execução dos testes envolve a participação de pessoas externas às empresas e, inclusive muitas vezes, de outros países.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] CRESPO, A. N.; JINO, M. Confiabilidade. In: DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. (Ed.). *Introdução ao Teste de Software*. 1. ed. Rio de Janeiro: Elsevier, 2007. cap. 13, p. 315–357.
- [2] PRESSMAN, R. S. *Engenharia de software: uma abordagem profissional*. 7. ed. Porto Alegre: McGraw Hill, 2011. 771 p.
- [3] PFLEEGER, S. L. *Engenharia de Software: teoria e prática*. 2. ed. São Paulo: Prentice Hall, 2004. 535 p.
- [4] SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 529 p.
- [5] RIZZI, P. *Uma investigação sobre a aplicação dos processos de teste nas empresas de desenvolvimento de Software associadas ao PoloSul*. 1–15 p. — (Trabalho de Conclusão de Curso em Ciência da Computação, Universidade de Passo Fundo).
- [6] KAGANER, E. et al. Managing the human cloud. *MIT Sloan Management Review*, v. 54, n. 2, p. 23–32, 2013.
- [7] CAPGEMINI; SOGETI; HP. *World Quality Report 2013-14*. [S.l.], 2014, 64 p.
- [8] MAO, K. et al. *A Survey of the Use of Crowdsourcing in Software Engineering*. London, 2015, 1–36 p.
- [9] OLIVEIRA, M. M. d. *Como fazer projetos, relatórios, monografias, dissertações e teses*. 5. ed. Rio de Janeiro: Elsevier, 2011. 232 p.
- [10] WAZLAWICK, R. S. *Metodologia de pesquisa para ciência da computação*. 1. ed. Rio de Janeiro: Elsevier, 2009. 159 p.
- [11] YIN, R. K. *Estudo de caso: planejamento e métodos*. 4. ed. Porto Alegre: Bookman, 2010. 248 p.
- [12] CARLI, E. *Gestão de mudanças aplicada a projetos: ferramentas de Change Management para unir PMO e CMO*. 1. ed. Rio de Janeiro: Brasport, 2015. 288 p.
- [13] International Organization for Standardization. *ISO 9000 Introduction and Support Package: Guidance on the Concept and Use of the Process Approach for management systems*. Genebra, 2008.

- [14] VALLE, R.; OLIVEIRA, S. B. (Ed.). *Análise e modelagem de processos de negócio: foco na notação bpmn (business process modeling notation)*. 1. ed. São Paulo: Atlas, 2013. 207 p.
- [15] MYERS, G. J. *The Art of Software Testing*. 1. ed. New York: Wiley, 1979. 177 p.
- [16] BARTIÉ, A. *Garantia da qualidade de software: adquirindo maturidade organizacional*. 1. ed. Rio de Janeiro: Elsevier, 2002. 291 p.
- [17] KOSCIANSKI, A.; SOARES, M. d. S. *Qualidade de software: aprenda as metodologias mais modernas para o desenvolvimento de software*. 2. ed. São Paulo: Novatec Editora, 2007. 395 p.
- [18] FABBRI, S. C. P. F.; VINCENZI, A. M. R.; MALDONADO, J. C. Teste Funcional. In: DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. (Ed.). *Introdução ao Teste de Software*. 1. ed. Rio de Janeiro: Elsevier, 2007. cap. 2, p. 9–25.
- [19] HOWE, J. The rise of crowdsourcing. *Wired magazine*, n. 14, p. 1–5, 2006.
- [20] PENG, X.; Ali Babar, M.; EBERT, C. Collaborative Software Development Platforms for Crowdsourcing. *IEEE Software*, v. 31, n. 2, p. 30–36, 2014.
- [21] STOL, K.-J.; FITZGERALD, B. Two's company, three's a crowd: a case study of crowdsourcing software development. In: *Proceedings of the 36th International Conference on Software Engineering - ICSE 2014*. New York, New York, USA: ACM Press, 2014. p. 187–198.
- [22] HOWE, J. *O Poder das Multidões*. 1. ed. Rio de Janeiro: Campus, 2009. 300 p.
- [23] PRIKLADNICKI, R. et al. Brazil software crowdsourcing: a first step in a multi-year study. In: *Proceedings of the 1st International Workshop on CrowdSourcing in Software Engineering - CSI-SE 2014*. New York: ACM Press, 2014. p. 1–4.
- [24] ALLAHBAKHS, M. et al. Quality Control in Crowdsourcing Systems: Issues and Directions. *IEEE Internet Computing*, v. 17, n. 2, p. 76–81, mar. 2013.
- [25] KEIMEL, C. et al. QualityCrowd — A framework for crowd-based quality evaluation. In: *2012 Picture Coding Symposium*. Krakow: IEEE, 2012. p. 245–248.
- [26] TSAI, W.-t.; WU, W.; HUHNS, M. N. Cloud-Based Software Crowdsourcing. *IEEE Internet Computing*, v. 18, n. 3, p. 78–83, maio 2014.
- [27] VUKOVIC, M. Crowdsourcing for Enterprises. In: *2009 World Conference on Services - I*. Los Angeles, CA: IEEE, 2009. p. 686–692.

- [28] LI, K. et al. Analysis of the Key Factors for Software Quality in Crowdsourcing Development: An Empirical Study on TopCoder.com. In: *2013 IEEE 37th Annual Computer Software and Applications Conference*. Kyoto: IEEE, 2013. p. 812–817.
- [29] USUI, Y.; MORISAKI, S. An Approach for Crowdsourcing Software Development. In: *Proceedings of IWSMMENSURA*. Nara, Japan: [s.n.], 2011. p. 32–33.
- [30] DOLSTRA, E.; VLIEGENDHART, R.; POUWELSE, J. Crowdsourcing GUI Tests. In: *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*. Luembourg: IEEE, 2013. p. 332–341.
- [31] CHEN, Z.; LUO, B. Quasi-crowdsourcing testing for educational projects. In: *Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014*. New York, New York, USA: ACM Press, 2014. p. 272–275.
- [32] Object Management Group. *Business Process Model and Notation (BPMN), version 2.0.2*. Needham, 2013.
- [33] GUEDES, G. T. A. *UML 2: uma abordagem prática*. 2. ed. São Paulo: Novatec, 2011. 484 p.
- [34] IEEE Std 829-2008. *IEEE Standard for Software and System Test Documentation*. New York, 2008.

## APÊNDICE A – PROJETO DE PESQUISA

Este projeto é responsável pelo planejamento da pesquisa a ser realizada no estudo de caso. Neste contexto, o presente projeto busca avaliar o uso do processo de teste proposto neste trabalho, levantando evidências que confirmem ou refutem as proposições elencadas. Adicionalmente, o projeto também apresenta a questão de estudo, que justifica a realização deste estudo de caso, além de apresentar também o tipo de estudo de caso e os critérios de seleção das unidades de análise, bem como a estratégia utilizada para interpretar os achados.

### QUESTÃO DE ESTUDO

Como o processo de teste se comporta ao ser utilizado em uma pequena empresa de desenvolvimento de software?

### PROPOSIÇÕES

1. As empresas são capazes de realizar testes funcionais utilizando o processo proposto;
2. O processo é genérico, ou seja, as empresas são capazes de integrar o processo de teste ao processo de desenvolvimento que utilizam;
3. Os testadores geograficamente dispersos são capazes de executar as tarefas de teste;
4. Crowdsourcing é uma abordagem viável para o teste funcional;
5. Crowdsourcing é uma abordagem barata de executar testes funcionais;
6. O resultado recebido ao final da tarefa de teste é relevante;
7. O idioma não é barreira para o uso do processo nas empresas;
8. O uso das plataformas não é barreira para o uso do processo nas empresas;
9. A proteção da propriedade intelectual não é barreira pra o uso do processo nas empresas;
10. O uso de reputação e preço são relevantes na escolha do testador;
11. O meio de pagamento e a segurança da transação não são barreiras para o uso do processo nas empresas;
12. O processo é suficientemente documentado para o entendimento por parte das empresas;

13. O processo permite o uso de qualquer técnica de teste funcional (análise de valor limite, partição de equivalência, dentre outras);
14. O processo é executado em um tempo adequado;
15. O processo flexibiliza a demanda de mão-de-obra de profissionais de teste, de acordo com a demanda da empresa.

## UNIDADES DE ANÁLISE

O estudo de caso será do tipo integrado (único), com duas unidades integradas de análise. Para a seleção das unidades de análise foram considerados os seguintes critérios:

- Empresa de desenvolvimento de software de pequeno porte (até 20 funcionários);
- Empresa de desenvolvimento de software que não realiza testes de maneira adequada;
- Empresa de desenvolvimento de software da cidade de Passo Fundo;
- Empresa de desenvolvimento de software que concorde em disponibilizar um software para execução do processo de teste e que tenha tempo disponível para acompanhar a execução do mesmo;
- Empresa de desenvolvimento de software que concorde em custear as despesas financeiras da execução do processo de teste (pagamento à plataforma crowdsourcing e aos testadores).

## VINCULAÇÃO DOS DADOS ÀS PROPOSIÇÕES E CRITÉRIOS PARA AS INTERPRETAÇÕES DOS ACHADOS

Como forma de vinculação dos dados às proposições e análise dos dados será utilizada a técnica de construção da explanação, onde serão descritos os procedimentos realizados em cada unidade de análise em que o estudo de caso será conduzido. Posteriormente cada proposição será avaliada, buscando confirmar ou refutar as afirmativas com base na análise dos dados coletados nas unidades de análise.

## APÊNDICE B – PROTOCOLO PARA O ESTUDO DE CASO

### VISÃO GERAL

O presente protocolo apresenta os procedimentos de campo para que o estudo de caso seja realizado. Além disso, o protocolo apresenta também as questões a serem respondidas, juntamente com suas prováveis fontes de evidência. Por fim, o protocolo apresenta um planejamento de organização do relatório do estudo de caso.

### PROCEDIMENTOS DE CAMPO

Em cada unidade de análise serão realizados os seguintes procedimentos de campo:

- Reunião de apresentação e planejamento das atividades;
- Levantamento da situação da empresa em relação ao teste;
- Apresentação do processo;
- Obtenção dos recursos necessários para a execução do processo;
- Integração do processo ao processo da empresa;
- Implantação e execução do processo;
- Acompanhamento da execução do processo, com reuniões após a conclusão de cada atividade;
- Reunião de conclusão dos trabalhos, guiada por roteiro de avaliação.

Observação: A coleta de dados ocorrerá de maneira contínua em todos os procedimentos anteriores.

### PROCEDIMENTOS PARA PROTEÇÃO DOS PARTICIPANTES (E DA EMPRESA)

Todos os elementos que identifiquem a empresa, como por exemplo seu nome, nome de seus funcionários, endereço, nome dos produtos que desenvolve. . . serão omitidos dos relatos. Da mesma forma a identidade dos testadores recrutados pelas plataformas crowdsourcing será preservada.

## QUESTÕES DE ESTUDO DE CASO

1. As empresas são capazes de realizar testes funcionais utilizando o processo proposto?
  - Fontes prováveis de evidência: documentação, registros em arquivo, entrevista, observação direta e observação participante.
2. O processo é genérico, ou seja, as empresas são capazes de integrar o processo de teste ao processo de desenvolvimento já em execução?
  - Fontes prováveis de evidência: documentação, registros em arquivo, observação direta e observação participante.
3. Os testadores geograficamente dispersos são capazes de executar as tarefas de teste?
  - Fontes prováveis de evidência: registros em arquivo e observação direta.
4. Crowdsourcing é uma abordagem viável para o teste funcional?
  - Fontes prováveis de evidência: documentação, registros em arquivo, entrevista, observação direta e observação participante.
5. Crowdsourcing é uma abordagem barata de executar testes funcionais?
  - Fontes prováveis de evidência: registros em arquivo, entrevista e observação participante.
6. O resultado recebido ao final da tarefa de teste é relevante?
  - Fontes prováveis de evidência: documentação, registros em arquivo, entrevista, observação direta e observação participante.
7. O idioma é barreira para o uso do processo nas empresas?
  - Fontes prováveis de evidência: entrevista, observação direta e observação participante.
8. O uso das plataformas crowdsourcing é barreira para o uso do processo nas empresas?
  - Fontes prováveis de evidência: entrevista, observação direta e observação participante.
9. A proteção da propriedade intelectual é barreira pra o uso do processo nas empresas?

- Fontes prováveis de evidência: entrevista, observação direta e observação participante.
10. O uso de reputação e preço são relevantes na escolha do testador?
- Fontes prováveis de evidência: entrevista, observação direta e observação participante.
11. O meio de pagamento e a segurança da transação são barreiras para o uso do processo nas empresas?
- Fontes prováveis de evidência: entrevista, observação direta e observação participante.
12. O processo é suficientemente documentado para o entendimento por parte das empresas?
- Fontes prováveis de evidência: documentação, registros em arquivo, entrevista, observação direta e observação participante.
13. O processo permite o uso de qualquer técnica de teste funcional (análise de valor limite, partição de equivalência, dentre outras)?
- Fontes prováveis de evidência: documentação, registros em arquivo, observação direta e observação participante.
14. O processo é executado em um tempo adequado?
- Fontes prováveis de evidência: documentação, registros em arquivo, entrevista, observação direta e observação participante.
15. O processo flexibiliza a demanda de mão de obra de profissionais de teste, de acordo com a demanda da empresa?
- Fontes prováveis de evidência: documentação, registros em arquivo, entrevista, observação direta e observação participante.

## GUIA DO RELATÓRIO DE ESTUDO DE CASO

1. Apresentação;
2. Descrição do projeto de pesquisa e do protocolo do estudo de caso;
3. Apresentação do estudo piloto;
4. Apresentação dos dados coletados em cada unidade de análise;
5. Análise e discussão dos dados coletados.

## APÊNDICE C – LEVANTAMENTO INICIAL

O seguinte questionário tem como objetivo levantar as informações da empresa, necessárias para estabelecer um panorama quanto à sua estrutura e quanto ao uso de testes.

1. Que tipos de produtos a empresa desenvolve?
  - São soluções sob demanda ou soluções “de prateleira”?
2. Tecnologias que utiliza:
  - Quanto ao software em si;
  - Tecnologias utilizadas para desenvolver o software:
    - IDEs;
    - Softwares para testes;
    - Banco de dados;
    - Sistemas de gestão;
3. Quanto à equipe:
  - Número de profissionais;
  - Organização dos profissionais na empresa (papéis, atribuições. . . );
4. Quanto ao processo de desenvolvimento de software:
  - A empresa segue algum processo “da literatura”?
  - Quais as atividades do processo?
  - Que tipo de documentação é realizada no processo?
5. Quanto ao teste:
  - Que níveis de teste são executados?
  - Que técnicas de teste são utilizadas?
  - Que atributos de teste são avaliados?
6. Quanto ao teste funcional:
  - É realizado?
  - De que forma é realizado?
  - Que técnicas são utilizadas?
  - Que softwares são utilizados nos testes?

## APÊNDICE D – ROTEIRO DE AVALIAÇÃO

O seguinte roteiro tem como objetivo avaliar a execução do processo na empresa. Para cada item do roteiro, considerar e relatar as percepções e considerações da empresa com a execução do processo, bem como sugerir melhorias:

- Em relação ao tempo de execução das atividades;
- Em relação ao resultado recebido no final do processo;
- Em relação ao processo de negociação com o testador;
- Em relação ao uso do inglês como idioma “padrão”;
- Em relação à habilidade dos testadores da multidão;
- Em relação ao uso da plataforma crowdsourcing;
- Em relação ao custo financeiro para a execução dos testes;
- Em relação à segurança do pagamento;
- Em relação à segurança da propriedade intelectual da empresa;
- Em relação à documentação para apoio do processo;
- Em relação à flexibilização da mão-de-obra de profissionais de teste;
- Em relação à facilidade de acesso ao website do processo (<http://www.softwarecrowdsourcing.com.br/cpft>);
- Utilizaria o processo na empresa? Por quê?
- Faria modificações no processo? Por quê? Quais?
- Realizaria outras atividades com a universidade?