



**PODER EXECUTIVO**  
**MINISTÉRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DO AMAZONAS**



**INSTITUTO DE COMPUTAÇÃO**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**LINHA DE PROCESSO DE SOFTWARE PARA APOIAR TESTES EM**  
**APLICAÇÕES MÓVEIS**

Kariny Marques de Oliveira

Manaus

2017



**PODER EXECUTIVO**  
**MINISTÉRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DO AMAZONAS**  
**INSTITUTO DE COMPUTAÇÃO**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**



**LINHA DE PROCESSO DE SOFTWARE PARA APOIAR TESTES EM**  
**APLICAÇÕES MÓVEIS**

Kariny Marques de Oliveira

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas, como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

Orientador: Arilo Claudio Dias Neto, DSc.

Manaus

2017

### Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Oliveira, Kariny Marques  
O48l Linha de processo de software para apoiar testes em aplicações  
móveis / Kariny Marques Oliveira. 2017  
115 f.: il.; 31 cm.

Orientador: Arilo Claudio Dias Neto  
Dissertação (Mestrado em Informática) - Universidade Federal do  
Amazonas.

1. Linha de processo de software. 2. Teste de software. 3.  
Aplicações móveis. 4. Processo de teste de software. I. Dias Neto,  
Arilo Claudio II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
INSTITUTO DE COMPUTAÇÃO



UFAM

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

## FOLHA DE APROVAÇÃO

"LINHA DE PROCESSO DE SOFTWARE PARA APOIAR TESTES EM  
APLICAÇÕES MÓVEIS"

**KARINY MARQUES DE OLIVEIRA**

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos  
Professores:

*Arilo Claudio Dias Neto*

Prof. Arilo Claudio Dias Neto - PRESIDENTE

*R. R. Carvalho*  
Prof. José Reginaldo Hughes Carvalho - MEMBRO INTERNO

*Moisés Gomes de Carvalho*  
Prof. Moisés Gomes de Carvalho - MEMBRO EXTERNO

Manaus, 25 de Setembro de 2017

*A Deus, a minha família e aos meus amigos.*

## **Agradecimentos**

A Deus por me dar saúde e me guiar nesse caminho, por todos os ensinamentos e por cuidar da minha família, sem Ele nada seria possível.

A meus pais, por me ensinar que a única forma de crescer é por meio do estudo e que as oportunidades são apresentadas apenas uma vez e devemos aproveitá-las.

A meu namorado, pelo seu amor e apoio incondicional, certamente sem seu apoio essa tarefa teria sido mais difícil, obrigado por ser meu suporte, amigo e companheiro em todo momento.

A minha sobrinha, que sempre que a visito irradia luz, alegria e ilumina meu dia.

Agradeço também ao professor Arilo, meu orientador, pelas sábias palavras e ensinamentos durante toda a caminhada, desde a graduação. Obrigada por ser meu guia durante os obstáculos e ajudar a superá-los, me mostrar a visão da pesquisa acadêmica e unir forças para a conclusão deste trabalho.

Aos professores Dr. José Reginaldo e Dr. Moisés Carvalho, por terem aceito ser parte da banca e pelas contribuições ao longo da pesquisa.

A todos meus amigos do grupo ExperTS, Awdren, Silvia, Renata, Karina, Allan, Larissa, Ingrid, Anderson, Oswald, Laiza, Josias, Erick e todos os que já culminaram, pelo seu apoio, tempo e amizade, pelas conversas e conhecimento compartilhado.

Aos professores Eduardo Souto, Tayana Conte, Bruno Gadelha e Thelma Colanzi pelos ensinamentos sobre pesquisa, conhecimentos e dicas compartilhadas.

Ao INDT pela experiência pesquisa / indústria, por meio da parceria com a UFAM e posteriormente como ambiente profissional.

A todos os participantes que fizeram parte do estudo experimental, obrigada por seu tempo e contribuição.

Aos professores e pessoal administrativo do Instituto de Computação pelo suporte durante o tempo do mestrado.

Finalmente, à CAPES pelo apoio financeiro ao longo do mestrado e todos que colaboraram com essa pesquisa.

Resumo de Dissertação apresentada à UFAM/AM como parte dos requisitos necessários para à obtenção do título de Mestre em Informática (M. Sc.)

## LINHA DE PROCESSO DE SOFTWARE PARA APOIAR TESTES EM APLICAÇÕES MÓVEIS

Kariny Marques de Oliveira

Setembro / 2017

Orientador: Arilo Claudio Dias Neto, DSc.

A evolução das aplicações móveis nos últimos anos tem resultado em um desafio constante para a área de Engenharia de Software. O exponencial crescimento desse mercado e da criticidade dos sistemas desenvolvidos impõem uma maior atenção a aspectos de qualidade dessas aplicações, como por exemplo qual técnica de teste aplicar a certos tipos de aplicações. O teste de software consome uma quantidade significativa de recursos do projeto e tempo do ciclo de desenvolvimento. Custos normais de teste variam de 40% a 85%, de acordo com a criticidade da aplicação. Sendo assim, qualquer iniciativa para reduzir o esforço, tempo ou custo do ciclo de desenvolvimento deve incluir o processo de testes, e este deve ser adaptado de acordo com as características da aplicação a ser testada. Como alternativa de apoio a este cenário, este trabalho propõe o desenvolvimento de uma Linha de Processo de Software (LPrS) de modo a simplificar o controle de variabilidades e possibilitar a derivação de novos processos de testes de aplicações móveis, auxiliando o gerente/líder de testes na seleção de elementos de teste (i.e. atividades, tipos de teste, nível de teste, responsáveis pelas atividades, fases e ferramentas) mais adequados, levando em consideração as especificidades de cada aplicação. Uma ferramenta intitulada MTPLT (*Mobile Testing Process Line Tool*) foi desenvolvida com o objetivo de oferecer o apoio ferramental necessário ao gerente/líder de teste, facilitando a manipulação de dados referentes à aplicação e auxílio na tomada de decisão a respeito do processo a ser adotado. A ferramenta MTPLT foi submetida à avaliação por especialistas da área de teste de software com o objetivo de verificar sua viabilidade de uso. Como resultado, foi observada uma tendência à aceitação quando observados os quesitos utilidade percebida e facilidade de uso. Em utilidade percebida foram obtidas 46 respostas positivas contra 4 respostas negativas. Ao analisar a facilidade de uso, foram obtidas 30 respostas positivas e apenas 3 negativas. Também foram realizadas análises avaliando perfis, tempos de experiência e papéis desempenhados. Participantes com experiência em atuação na indústria e em pesquisas científicas apresentaram considerações a respeito da ferramenta quanto a apresentação de conteúdo e adição de funcionalidades.

Palavras-chave: linha de processo de software, teste de software, aplicações móveis.

Abstract of Thesis presented to UFAM/AM as a partial fulfillment of the requirements for the degree of Master of Science (M. Sc.)

## SOFTWARE PROCESS LINE TO SUPPORT MOBILE APPLICATION TESTING

Kariny Marques de Oliveira

September / 2017

Advisor: Arilo Claudio Dias Neto, DSc.

The evolution of mobile applications in recent years has resulted in a constant challenge for the Software Engineering area. The exponential growth of this market and the criticality of the developed systems impose a greater attention to quality aspects of these applications, such as which test technique to apply to certain types of applications. Software testing consumes a significant amount of project resources and development cycle time. Normal test costs range from 40% to 85%, according to the application's criticality. Therefore, any initiative to reduce the effort, time, or cost of the development cycle should include the testing process, and this should be tailored according to the characteristics of the application under test. As an alternative to support this scenario, this work proposes the development of a Software Process Line (SPrL) in order to simplify the control of variabilities and enable the derivation of new mobile application testing processes, assisting the test manager/leader in the selection of test elements (i.e. activities, test types, test level, responsible for activities, phases and tools), taking into account the specificities of each application. The tool called MTPLT (Mobile Testing Process Line Tool) was developed with the aim of providing the necessary tooling support to the manager/test leader, facilitating the manipulation of data related to the application and aid in decision making regarding the process to be adopted. The MTPLT was evaluated by experts in software testing in order to verify its feasibility in industry environments. As a result, it was observed a tendency to acceptance when observed the requirements perceived utility and ease of use. In the perceived utility, 46 positive responses were obtained against 4 negative responses. When analyzing ease of use, 30 positive and only 3 negative responses were obtained. Analyses were carried out evaluating profiles, experience times and roles performed. Most of the participants with previous experience on the industry and researchers presented considerations regarding the presentation of content and the addition of features.

*Keywords: Software Process Line, Software Testing, Mobile Applications.*

## Sumário

LISTA DE ABREVIATURAS E SIGLAS .....	12
LISTA DE FIGURAS .....	13
LISTA DE TABELAS .....	15
1. Introdução .....	17
1.1 Contexto e Motivação .....	17
1.2 Hipótese.....	19
1.3 Objetivos .....	19
1.3.1 Objetivo Geral .....	19
1.3.2 Objetivos Específicos .....	20
1.4 Metodologia de pesquisa.....	20
1.4.1 Fase de concepção .....	20
1.4.2 Fase de avaliação.....	21
1.5 Estrutura do Documento.....	21
2. REFERENCIAL TEÓRICO .....	23
2.1. Engenharia de Processo de Software.....	23
2.2. Linha de Processo de Software .....	27
2.2.1. Engenharia de Domínio (ED).....	28
2.2.2. Engenharia de Aplicação (EA).....	28
2.3. Processo de Teste de Software .....	29
2.4. Mapeamento Sistemático sobre Processos de Teste de Software .....	32
2.4.1. Protocolo do mapeamento sistemático .....	33
2.4.1.1. Questões de pesquisa.....	33
2.4.1.2. Identificação e Seleção de Estudo Primários.....	34
2.4.1.3. Formulação de Extração de Dados.....	34
2.4.2. Execução do Estudo do Mapeamento Sistemático.....	34

2.4.3.	Papéis que participam do processo.....	35
2.4.4.	Fases componentes.....	36
2.4.5.	Tipos de teste utilizados .....	37
2.4.6.	Considerações do mapeamento .....	37
2.5.	Extensão de mapeamento sistemático sobre Linhas de Processo de Software.....	38
2.5.1.	Método de pesquisa .....	38
2.5.1.1.	Planejamento .....	38
2.5.1.2.	Execução .....	40
2.5.2.	Extração dos dados e resultados .....	43
2.6.	Teste de Aplicações Móveis.....	51
2.7.	Trabalhos Relacionados .....	54
3.	MTPL: linha de processo de software para teste de aplicações móveis.....	56
3.1.	Introdução .....	56
3.2.	Conteúdo da MTPL.....	56
3.3.	Processo da Linha de Processo de Software .....	59
3.3.1.	Fase de planejamento .....	59
3.3.2.	Design & Implementação.....	60
3.3.3.	Execução .....	66
3.3.4.	Análise e encerramento .....	69
3.3.5.	Processos para teste de usabilidade .....	69
3.4.	Apoio computacional à linha proposta.....	76
3.4.1.	Funcionamento da MTPLT .....	76
4.	Avaliação da linha de processos proposta.....	82
4.1.	Definição do estudo de viabilidade .....	82
4.1.1.	Propósito.....	82
4.1.2.	Perspectiva .....	82
4.1.3.	Objetivo específicos .....	82

4.1.4.	Questões e métricas .....	82
4.2.	Planejamento do estudo.....	83
4.2.1.	Seleção dos participantes .....	83
4.2.2.	Seleção de grupos.....	84
4.3.	Execução do estudo.....	84
4.4.	Resultados do estudo.....	85
4.4.1.	Caracterização dos participantes .....	85
4.4.2.	Análise quantitativa.....	87
4.4.2.1.	Facilidade de uso.....	87
4.4.2.2.	Utilidade percebida .....	89
4.4.2.3.	Tendências.....	91
4.4.3.	Análise qualitativa.....	94
4.5.	Ameaças à validade .....	98
4.5.1.	Validade interna .....	98
4.5.2.	Validade externa.....	98
4.5.3.	Validade de construção .....	98
4.5.4.	Validade de conclusão.....	98
4.6.	Considerações finais.....	98
5.	CONCLUSÕES E TRABALHOS FUTUROS .....	100
5.1.	Considerações finais.....	100
5.2.	Contribuições da pesquisa.....	101
5.3.	Limitações .....	102
5.4.	Trabalhos futuros.....	102
	Referências Bibliográficas .....	103

## LISTA DE ABREVIATURAS E SIGLAS

BPM	<i>Business Process Management</i>
CMMI	<i>Capability Maturity Model Integration</i>
EPF	<i>Eclipse Process Framework</i>
ISO	<i>International Organization for Standardization</i>
LPrS	<i>Linha de Processo de Software</i>
LPS	<i>Linha de Produto de Software</i>
MA-MPS	Método de avaliação para Melhoria do Processo de Software
MPS.BR	Melhoria de Processo do Software Brasileiro
MTPL	<i>Mobile Testing Process Line</i>
MTPLT	<i>Mobile Testing Process Line Tool</i>
OMG	<i>Object Management Group</i>
PL	<i>Product Line</i>
SCAMPi	<i>Standard CMMI Appraisal Method for Process Improvement</i>
SPEM	<i>Software Process Engineering Metamodel</i>
SPL	<i>Software Product Line</i>
SPrL	<i>Software Process Line</i>
UML	<i>Unified Modeling Language</i>

## LISTA DE FIGURAS

<b>Figura 1.</b> Metodologia de pesquisa. ....	20
<b>Figura 2.</b> As camadas da Engenharia de Software. ....	23
<b>Figura 3.</b> Áreas de conhecimento da engenharia de processo de software ....	24
<b>Figura 4.</b> Elementos básicos do SPEM ....	26
<b>Figura 5.</b> Engenharia de processos para LPrS.....	29
<b>Figura 6.</b> Modelo em camadas do processo de teste ....	29
<b>Figura 7.</b> Processo de Planejamento de Teste. ....	30
<b>Figura 8.</b> Processo Monitoramento e Controle.....	30
<b>Figura 9.</b> Processo de encerramento de testes. ....	31
<b>Figura 10.</b> Processo de Projeto e Implementação ....	31
<b>Figura 11.</b> Processo de Definição & Manutenção de Ambiente de Teste.....	31
<b>Figura 12.</b> Processo de Execução dos Testes. ....	32
<b>Figura 13.</b> Processo de Documentação de Incidentes. ....	32
<b>Figura 14.</b> Publicações por ano ....	35
<b>Figura 15.</b> Principais contribuições identificadas. ....	43
<b>Figura 16.</b> Tipos de variabilidade encontradas.....	44
<b>Figura 17.</b> Tipos de dependências encontradas.....	45
<b>Figura 18.</b> Elementos do processo ....	46
<b>Figura 19.</b> Linguagens de modelagem identificadas.....	47
<b>Figura 20.</b> Modelos e padrões de qualidade de processo identificados.....	48
<b>Figura 21.</b> Paradigmas de engenharia de software.....	49
<b>Figura 22.</b> Conceitos de linha de processo utilizados ....	49
<b>Figura 23.</b> Ferramentas de suporte.....	50
<b>Figura 24.</b> Métodos de validação utilizados.....	51
<b>Figura 25.</b> Ciclo de vida de teste de software ....	59
<b>Figura 26.</b> Fase de design & implementação ....	61
<b>Figura 27.</b> Fase de execução ....	66
<b>Figura 28.</b> Processo de preparação de testes de usabilidade. ....	70
<b>Figura 29.</b> Processo de recrutamento de participantes para testes de usabilidade.....	70
<b>Figura 30.</b> Design & Implementação para teste de Usabilidade ....	71

<b>Figura 31.</b> Execução para teste de Usabilidade.....	75
<b>Figura 32.</b> Tela selecionar estratégias de teste .....	77
<b>Figura 33.</b> Tela de selecionar abordagens de teste .....	78
<b>Figura 34.</b> Tela seleção testes automatizados.....	78
<b>Figura 35.</b> Plataformas de teste .....	79
<b>Figura 36.</b> Tela de seleção de ambiente de testes.....	80
<b>Figura 37.</b> Tela de relatório da aplicação MTPLT.....	80
<b>Figura 38.</b> Papel desempenhado por participantes do experimento .....	85
<b>Figura 39.</b> Conhecimentos prévios do Grupo A. ....	86
<b>Figura 40.</b> Conhecimentos prévios do Grupo B.....	86
<b>Figura 41.</b> Tempo de experiência em teste de aplicações móveis.....	87
<b>Figura 42.</b> Facilidade de uso do Grupo A .....	88
<b>Figura 43.</b> Facilidade de uso do Grupo B. ....	89
<b>Figura 44.</b> Utilidade percebida do Grupo A.....	90
<b>Figura 45.</b> Utilidade percebida do Grupo B. ....	90
<b>Figura 46.</b> Interpretação dos resultados.....	92
<b>Figura 47.</b> Modelo de características da MTPL .....	113
<b>Figura 48.</b> Restrições da MTPL .....	113
<b>Figura 49.</b> Diagrama de características da MTPL.....	114
<b>Figura 50.</b> Formulário de caracterização - Papel.....	118
<b>Figura 51.</b> Formulário de caracterização - nível de conhecimento .....	118
<b>Figura 52.</b> Formulário de caracterização - tempo de experiência .....	118
<b>Figura 53.</b> Realização da tarefa específica do grupo.....	119
<b>Figura 54.</b> Perguntas de avaliação da ferramenta.....	119
<b>Figura 55.</b> Campo de críticas e sugestões da ferramenta .....	119

## LISTA DE TABELAS

<b>Tabela 1.</b> Objetivo do mapeamento sistemático segundo o paradigma GQM. ....	33
<b>Tabela 2.</b> Formulário de extração de dados.....	34
<b>Tabela 3.</b> Número de artigos selecionados por fase .....	35
<b>Tabela 4.</b> Questões de pesquisa.....	39
<b>Tabela 5.</b> Critérios de inclusão .....	39
<b>Tabela 6.</b> Critérios de exclusão .....	40
<b>Tabela 7.</b> Strings de busca.....	41
<b>Tabela 8.</b> Strings da IEEE Xplore .....	41
<b>Tabela 8</b> (Conclusão) .....	42
<b>Tabela 9.</b> Atividade selecionar itens e características de teste .....	62
<b>Tabela 10.</b> Atividade descrever estratégias para o projeto de testes .....	62
<b>Tabela 11.</b> Atividade identificar casos de teste e procedimentos .....	63
<b>Tabela 11</b> (Continuação) .....	64
<b>Tabela 12.</b> Atividade priorizar casos de teste e procedimentos.....	64
<b>Tabela 13.</b> Atividade especificar critérios de aprovação / rejeição .....	65
<b>Tabela 14.</b> Atividade gerar scripts.....	65
<b>Tabela 15.</b> Atividade documentar projeto de testes.....	65
<b>Tabela 16.</b> Atividade configurar ambiente de testes .....	67
<b>Tabela 17.</b> Atividade executar procedimentos de teste .....	67
<b>Tabela 18.</b> Comparar resultados dos testes.....	68
<b>Tabela 19.</b> Atividade analisar critério de parada.....	68
<b>Tabela 20.</b> Atividade documentar execução dos testes .....	69
<b>Tabela 21.</b> Atividade selecionar itens e características de teste - Usabilidade.....	71
<b>Tabela 22.</b> Atividade descrever estratégias de teste - Usabilidade.....	72
<b>Tabela 23.</b> Atividade definir cenários, tarefas e sessões .....	72
<b>Tabela 24.</b> Atividade definir atributos, métricas e abordagens de coleta de dados .....	72
<b>Tabela 24</b> (Continuação) .....	73
<b>Tabela 25.</b> Atividade preparar material de apoio .....	73
<b>Tabela 26.</b> Atividade selecionar participantes.....	73
<b>Tabela 26.</b> (Continuação) .....	74

<b>Tabela 27.</b> Atividade documentar projeto de testes.....	74
<b>Tabela 28.</b> Atividade configurar ambiente de execução - usabilidade .....	75
<b>Tabela 29.</b> Atividade executar testes - usabilidade .....	75
<b>Tabela 30.</b> Atividade armazenar dados dos testes .....	75
<b>Tabela 30.</b> (Continuação) .....	76
<b>Tabela 31.</b> Tendência da utilidade percebida .....	92
<b>Tabela 32.</b> Tendência da facilidade de uso.....	93
<b>Tabela 33.</b> Sumário do participante 1.....	94
<b>Tabela 34.</b> Sumário do participante 2.....	95
<b>Tabela 35.</b> Sumário do participante 3.....	95
<b>Tabela 36.</b> Sumário do participante 4.....	96
<b>Tabela 37.</b> Sumário do participante 5.....	96
<b>Tabela 38.</b> Sumário do participante 6.....	97

# 1. INTRODUÇÃO

## 1.1 Contexto e Motivação

Uma aplicação móvel é um software projetado para executar em *smartphones*, *tablets* e outros dispositivos móveis que pode ou não considerar informações contextuais de entrada (CHEN e KOTZ, 2000). Inicialmente, as aplicações móveis foram desenvolvidas especialmente para o setor de entretenimento, porém agora estão sendo aplicadas em domínios mais críticos, tais como Varejo (comércio baseado na localização do cliente), Mídia (revistas e jornais 100% digitais), Viagens (reservas, *check-ins*, mapas, ofertas, etc.), Educação (*tablets* e aplicativos em salas de aula), Saúde (registro de paciente, monitoramento de pessoas, notas de médicos, etc.), Finanças (aplicativos em tempo real para negociação e análise de portfólio) e Sociais (jogos e mídias sociais) (DELAMARO et al., 2016). Ao mesmo tempo em que as aplicações móveis estão se tornando cada vez maiores e mais complexas, a exigência de qualidade nestes produtos também tem aumentado. No entanto, como já consolidado na Engenharia de Software, a qualidade de um produto de software também depende da qualidade dos processos de desenvolvimento adotados para construí-lo (CUGOLA e GHEZZI, 1998; FUGGETA, 2000; OSTERWEIL, 1987).

Um processo de software pode ser definido como um conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos necessários para conceber, desenvolver, implantar e manter um produto de software (FUGGETA, 2000). No entanto, a diversidade de organizações, projetos, suas características e objetivos torna a tarefa de definição de processos de software não trivial. Há uma grande quantidade de material indicando as melhores práticas a serem seguidas, mas cada organização deve definir seus processos levando em consideração não só essas informações, mas também suas próprias características (BERTOLLO et al., 2006). Devido ao grande número de processos oriundos dessas derivações, algumas técnicas têm sido utilizadas para reduzir o esforço de derivação e definição dos processos, dentre elas a reutilização de processos. Utilizando conceitos semelhantes à reutilização de produtos, a reutilização de processos tem o objetivo de definir processos baseando-se em um processo padrão, adicionando especificidades do projeto, aplicação ou organização.

O processo de teste possui atividades, fases, ferramentas, tipos de teste e responsáveis pela execução das tarefas. Todos esses elementos podem apresentar variações dependendo do domínio da aplicação a ser testada. Entende-se por domínio da aplicação a categoria ou meio ao qual é destinada. Os testes de aplicações móveis diferenciam-se de testes de aplicações tradicionais por possuírem alguns requisitos específicos (DELAMARO et al., 2016), são eles:

- Potencial interação com outras aplicações a partir de fontes variadas;
- Conectividade com uma rede móvel que pode variar em velocidade, confiabilidade e segurança;
- Diversificados métodos de entrada de dados que exigem um certo nível de proficiência, tais como acelerômetro, tela sensível ao toque (*touch screen*), teclados reais e/ou virtuais, sistema de posicionamento global (GPS), microfone, uma ou mais câmeras e vários protocolos de comunicação em rede.
- Sensibilidade ao contexto no qual estão inseridas (ex.: sensores de som, luzes, movimentos, tipo de conexão e localização) para funcionarem de forma adequada.
- Diversidade de famílias de hardware e plataformas de software nas quais uma aplicação móvel precisa funcionar, com diferentes sistemas operacionais, plano de telefonia e diferentes componentes de hardware e sensores que auxiliam na entrada de dados, de forma que dois dispositivos distintos executando a mesma aplicação em um mesmo ambiente podem computar diferentes resultados;
- Segurança, visto que as plataformas móveis são abertas, permitindo a instalação de novas aplicações “*malware*” que podem afetar a operação global do dispositivo, incluindo a transmissão oculta de dados locais por uma aplicação.
- Interface de usuário deve estar aderente a diretrizes desenvolvidas pela plataforma. Além disso, dispositivos móveis contêm tamanho de tela limitado e por isso a quantidade de informação que pode ser exibida é limitada e a resolução de dispositivos móveis é reduzida em relação à resolução de computadores *desktop*, resultando em imagens com mais baixa qualidade;
- Limitação no consumo de energia em comparação a *laptops* e computadores *desktops*. Assim, o uso de recursos do dispositivo móvel deve ser otimizado de modo a evitar a degradação do desempenho e funcionamento incorreto da aplicação;
- Novas linguagens de programação foram projetadas para suportar as características diferenciadas dessas aplicações, como mobilidade, gerenciamento de consumo de recursos e manipulação de novos elementos de interface gráfica;
- Complexidade dos testes, visto que as características citadas anteriormente tornam o processo de testes ainda mais desafiador e com características próprias, quando comparados aos testes de outras plataformas de software. Dessa forma, técnicas tradicionais de teste precisam ser revisadas de forma a serem aplicadas à nova plataforma de desenvolvimento de softwares para dispositivos móveis.

Tendo em vista que a qualidade das aplicações começa inicialmente pela definição do processo a ser adotado, torna-se necessário realizar uma definição adequada a respeito desse processo. Analisando os fatores mencionados anteriormente, a diversidade de características torna árdua a tarefa de definição do processo adequado. Dessa forma, uma das formas de gerenciar a grande quantidade de processos resultantes dessa instanciação é a adoção de Linhas de Processo de Software (LPrS). Segundo (TEIXEIRA e WERNER, 2014), uma Linha de Processo de Software é um conjunto de processos de software que compartilham um conjunto de características comuns e variáveis, e são desenvolvidos a partir de artefatos, que podem ser reutilizados e combinados entre si, segundo regras de composição e recorte, para compor e adaptar processos de software.

A utilização da abordagem de Linha de Processo de Software (LPrS) para derivação de processos adaptados em determinados domínios de aplicação pode gerar diversas vantagens como aumento da previsibilidade, redução de custo e tempo, e redução de riscos por meio da reutilização de elementos de processos de software. Com a melhoria e a rápida adaptação de processos aos diversos cenários é possível aumentar a qualidade dos produtos de software e acelerar seu desenvolvimento. Entre as principais tarefas dessa abordagem está a gerência e a adaptação de variabilidades em famílias de processos de software.

O conceito de LPrS foi utilizado devido a grande variedade de tipos de teste, ferramentas e abordagens utilizadas, dessa forma há um gerenciamento de aspectos variáveis relacionando-os da melhor forma para aumentar a qualidade da aplicação desenvolvida.

## **1.2 Hipótese**

A hipótese definida para este trabalho considera o seguinte cenário:

Uma ferramenta de apoio à definição de processos de teste de aplicação móveis a partir de conceitos de LPrS contribui para a melhoria da qualidade da aplicação na definição do processo de teste a ser utilizado.

## **1.3 Objetivos**

Nesta seção serão apresentados os objetivos desta pesquisa.

### **1.3.1 Objetivo Geral**

Definir e avaliar experimentalmente uma ferramenta baseado em linha de processos de software para instanciação de processos de teste para aplicações móveis visando apoiar o gerente de testes na indicação de atividades, técnicas, artefatos e ferramentas relacionadas às fases de projeto e execução dos testes de acordo com o domínio da aplicação móvel a ser testada.

### 1.3.2 Objetivos Específicos

- Apresentar um modelo de características (*feature model*) com ferramentas, atividades, tipos de teste e responsáveis envolvidos, extraídos de um mapeamento sistemático e trabalhos relacionados.
- Definir uma Linha de Processo de Software para apoiar o teste de aplicações móveis.
- Prover uma ferramenta que utilizando a linha de processo de software definida e a partir da análise de questões respondidas pelo gerente de testes indica o processo mais adequado para o tipo de aplicação.

## 1.4 Metodologia de pesquisa

A metodologia de pesquisa utilizada neste trabalho é composta por duas fases: **concepção**, onde o foco é a definição e construção da ferramenta MTPLT (*Mobile Testing Process Line Tool*), e; **avaliação**, focada em avaliar a utilização da ferramenta. Na Figura 1 são apresentadas as etapas dentro de cada fase, nas seções 1.4.1 (concepção) e 1.4.2 (avaliação) as etapas são detalhadas.

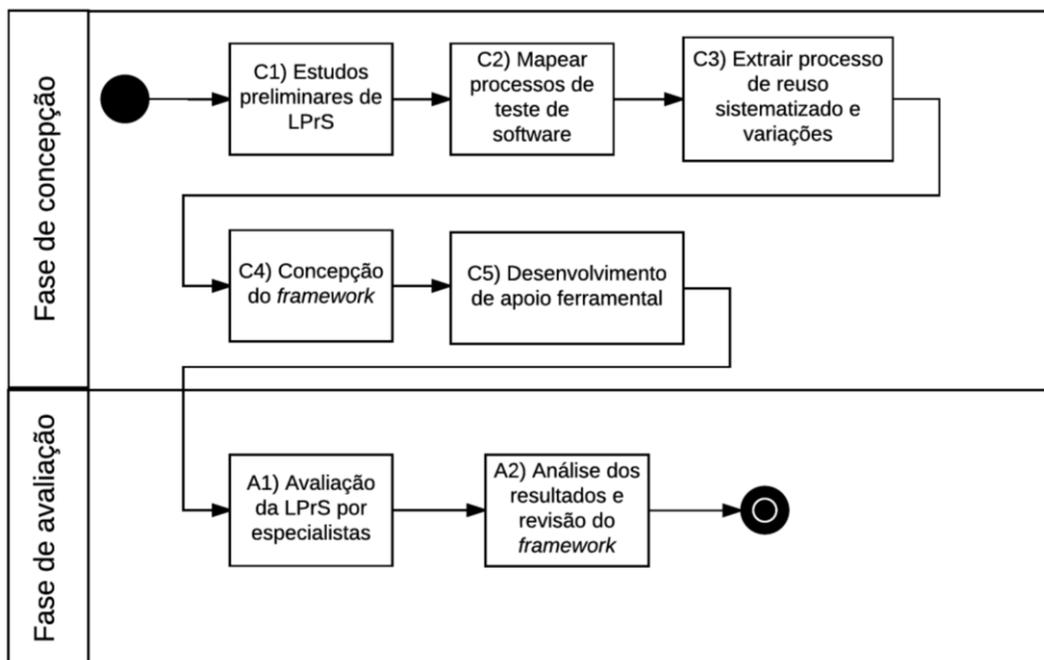


Figura 1. Metodologia de pesquisa.

### 1.4.1 Fase de concepção

- C1) Estudos preliminares de LPrS. Foi realizada uma busca informal na literatura com o objetivo de capacitar o pesquisador para torná-lo apto à realização deste trabalho. O trabalho envolve a compreensão dos processos de teste de software, incluindo tarefas,

atividades, ferramentas e papéis envolvidos, desde os processos clássicos aplicados no mercado, até os processos propostos para teste de aplicações móveis. O pesquisador também deve dominar a atividade de um engenheiro de processo de software, que é responsável pela customização dos processos de engenharia de software. O estudo preliminar pode ser visto na seção 2.2.

- **C2)** Foi realizado um mapeamento sistemático sobre os processos de teste de software, seguindo os princípios de uma revisão sistemática da literatura (KITCHENHAM e CHARTERS, 2007). O protocolo definido, os procedimentos de execução e a análise dos resultados do mapeamento sistemático podem ser vistos na seção 2.4.
- **C3)** Foi extraído o processo de reuso sistematizado (processo base) e as variações do mesmo, a partir dos resultados do mapeamento sistemático em conjunto com a ISO/IEEE 29119-2. Foram definidos pontos de variação apresentados na seção 3.2 e o modelo de características (*feature model*) disponível no Apêndice C.
- **C4)** Concepção da ferramenta. A partir dos resultados das atividades anteriores a ferramenta foi concebido, contendo os processos de teste possíveis com a combinação de características do modelo de características, disponíveis no Capítulo 3.
- **C5)** Após a concepção foi desenvolvido o apoio ferramental que utiliza a ferramenta concebido, baseado em questionário o usuário responde a uma série de perguntas e tem como resultado a indicação de um processo a ser seguido para testar a aplicação móvel especificada, disponível na seção 3.5.

#### **1.4.2 Fase de avaliação**

- **A1)** A avaliação da ferramenta foi realizada utilizando um *survey* com especialistas na indústria e academia, tendo em vista que a implantação e avaliação de uma LPrS em uma empresa demanda bastante tempo, por fatores como treinamento, resistência a novas tecnologias e migração de dados, disponível no Capítulo 4.
- **A2)** Após a avaliação os resultados apresentados pelos especialistas foram analisados e identificados pontos de melhorias na ferramenta, a partir desses pontos foi realizada uma revisão da ferramenta acrescentando as possíveis melhorias.

### **1.5 Estrutura do Documento**

Este Capítulo 1 apresentou uma visão geral da proposta de trabalho contextualizando o leitor sobre a relação com algumas áreas da engenharia de software. Apresentou a motivação, objetivo geral e específico, assim como a estruturação do processo de trabalho.

O Capítulo 2 apresenta o referencial teórico, focando os temas de processo de software, notação para modelagem de processos de software e teste de aplicações móveis.

O Capítulo 3 descreve a concepção da linha de processo e ferramenta de apoio.

O Capítulo 4 avalia a proposta do trabalho com especialistas da área.

O Capítulo 5 apresenta as conclusões e trabalhos futuros.

Os apêndices A até F apresentam o protocolo de pesquisa, componentes da linha e os formulários de avaliação da linha.

## 2. REFERENCIAL TEÓRICO

Este capítulo descreve alguns dos conceitos básicos relacionados a esta pesquisa, como engenharia de processos de software, linha de processo de software, processo de teste de software, teste de aplicações móveis e os trabalhos relacionados ao tema desta pesquisa.

### 2.1. Engenharia de Processo de Software

Processo de software é um conjunto de atividades uniformizadas a serem aplicadas sistematicamente que se encontram agrupadas em fases, cada uma das quais com os seus intervenientes com responsabilidades, que possui diversas entradas e produz diversas saídas. (SOMMERVILLE, 2015)

De acordo com Pressman (PRESSMAN, 2009), a fundamentação da engenharia de software é a camada de processo. O processo define um framework que deve ser estabelecido para entrega efetiva de uma tecnologia de engenharia de software (veja **Figura 2**).



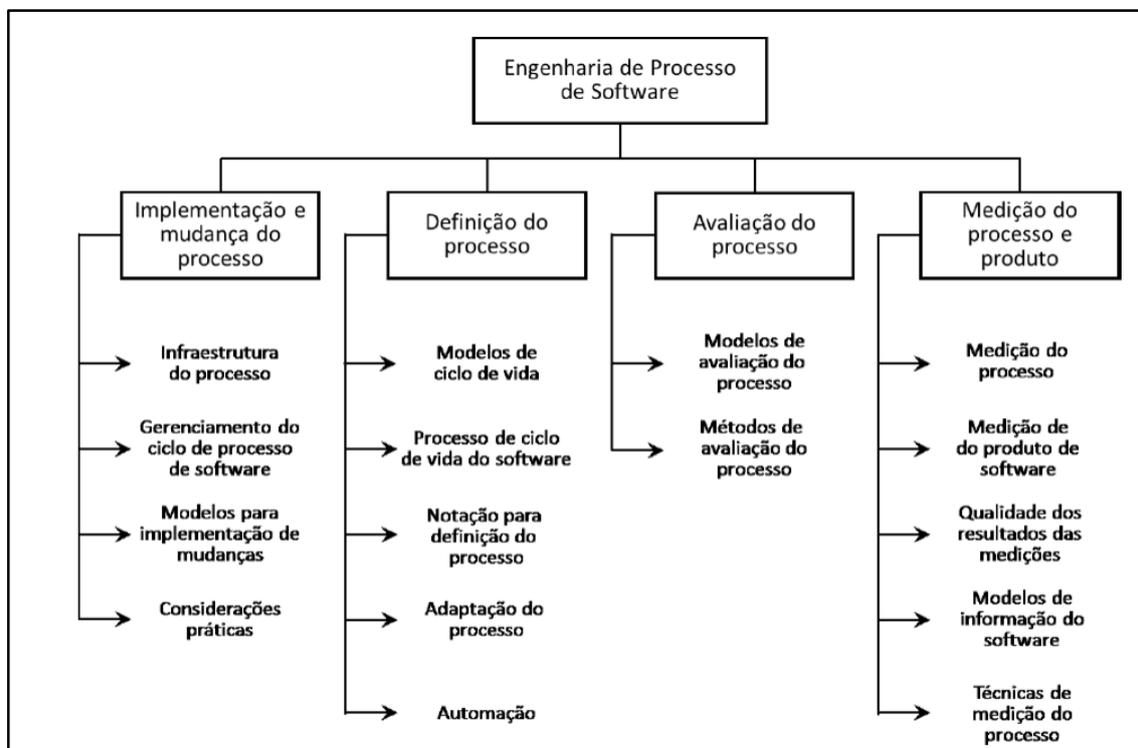
**Figura 2.** As camadas da Engenharia de Software. Adaptado de (PRESSMAN, 2009)

Um processo de teste de software é um conjunto de fases, atividades, papéis e ferramentas de apoio que são utilizadas quando algum produto está sendo avaliado:

- Fase: utilizada para alcançar um objetivo e é aplicada independentemente do domínio da aplicação, tamanho do projeto, esforço ou grau de rigor com que a engenharia de software está sendo aplicada. Uma fase envolve uma série de atividades.
- Atividade: tarefa focada em um objetivo menor, mas bem definido que produz um resultado tangível que pode ser associada com papéis e ferramentas de suporte.
- Papel (ex., testador): responsável pela execução de uma atividade.
- Ferramenta de suporte (ex., *TestLink*): ferramenta utilizada para apoiar as atividades do processo.

A engenharia de processo de software é responsável pela definição, modelagem, avaliação, medição, gestão, adaptação e melhoria do processo de ciclo de vida do software. O

*SWEBOK – A Guide to The Software Engineering Body of Knowledge* (IEEE, 2004) estruturou as áreas de conhecimento necessárias para a elaboração de um processo de engenharia de software, aqui referenciado como Engenharia de Processo de Software e apresentado na Figura 3.



**Figura 3.** Áreas de conhecimento da engenharia de processo de software, adaptado de (IEEE, 2004)

A primeira subárea de conhecimento (Implementação e Mudança no Processo) é dedicada à avaliação e descrição das mudanças na organização e seu impacto na infraestrutura, atividades, modelos e outras considerações práticas (IEEE, 2004). Na segunda subárea (Definição do Processo), enumeram-se os conhecimentos necessários para definição do processo, dos modelos de ciclo de vida, diagramação, adaptação de ferramentas que suportem o controle da execução do processo. Já na terceira subárea (Avaliação do Processo) está a avaliação baseada em modelos de referência para a melhoria de processos, como o CMMi-Dev (CMMI, 2006) e o MR-MPS (MPS.BR, 2012); e os métodos de avaliação, com procedimentos estruturados que gerem uma avaliação quantitativa ou capacitiva, como o SCAMPI (SCAMPI, 2011) e MA-MPS (MPS.BR, 2012). Por fim, na última subárea são apresentados os conhecimentos relativos à medição (Medição do Processo e Produto).

A necessidade de customizar processo de software foi motivada pela crescente demanda pela definição de melhorias contínuas para promover o desenvolvimento de software de qualidade com baixo custo. A adaptação de um modelo de processo de software é uma tarefa complexa e dispendiosa. Antes de realizar adaptações nos processos, faz-se necessária a análise e concepção das mudanças, atividade que requer muito tempo (ARMBRUST et al., 2008). Para elaborar,

estabelecer e treinar as equipes no processo de software, faz-se necessário uma notação que facilite a comunicação e a disseminação do processo.

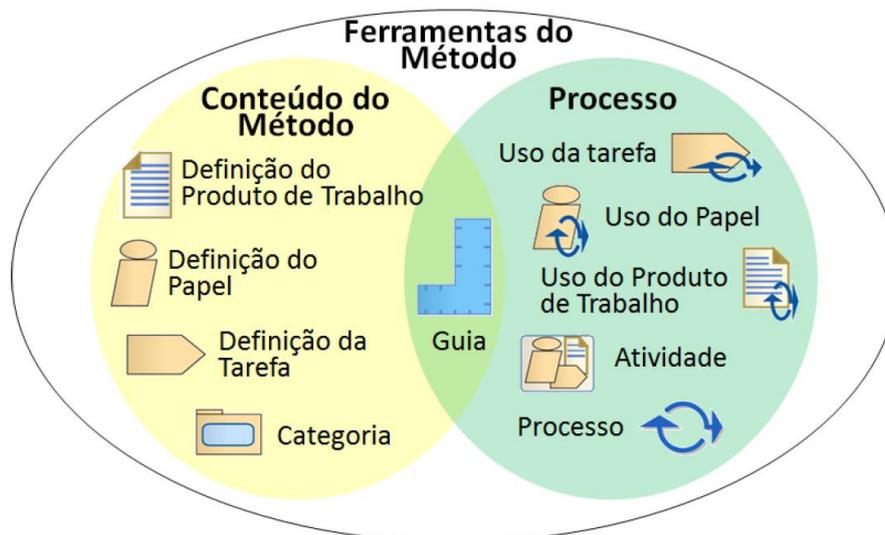
Existem propostas de linguagens para a modelagem de processos de software, porém diversas organizações utilizam simplesmente um diagrama de atividades ou fluxograma. Trabalhos anteriores realizaram a comparação entre os requisitos suportados entre as principais propostas de abordagens para a modelagem de processos, as quais estão avaliadas em diferentes perspectivas. Estas abordagens de representação são predominantemente baseadas em UML (*Unified Modeling Language*) e possuem uma riqueza semântica semelhante (BENDRAOU et al., 2010):

- A abordagem de DiNitto's e a Promenade são resultantes da especialização de um *framework* orientado a objetos, com uma camada de modelagem UML (DI NITTO et al., 2002; FRANCH et al., 1997);
- A abordagem de Chou's é resultante de diagramas de alto nível baseados na UML e uma linguagem adicional de baixo nível para o controle e execução do processo (CHOU, S, 2002);
- UML4SPM busca complementar o meta-nível, com suporte a tratamento a exceções e mecanismos para a execução (BENDRAOU et al., 2007);
- *Software Process Components* é uma abordagem simplificada que define basicamente três tipos de componentes: concretos, que não suportam variabilidades; abstratos, que são os pontos de variabilidades que serão substituídos por componentes variantes; e, opcionais. Estes componentes são ligados com associações do tipo: início-fim, início-início, fim-fim, fim-início (BARRETO et al., 2010);
- SPEM - *Software Process Engineering Metamodel* (versão 1.0 e 2.0) que possui origens tanto na forma do MOF *Metamodel* quanto nos perfis da UML (OMG, 2008). SPEM é o padrão da *Object Management Group* - OMG destinado à modelagem de processos de software, cuja versão 2.0 foi disponibilizada em 2008. Sua estrutura favorece a reutilização, pois separa elementos e aspectos entre o método de desenvolvimento e a instância de um processo em particular. A versão 2.0 inclui controles básicos de variabilidade como o uso da tarefa e o uso do produto de trabalho.
- vSPEM abordagem que complementa as variabilidades do SPEM versão 2.0. Permite a definição de pontos de variação e variantes, inclusive dentro do conteúdo do processo, possibilitando a definição genérica de um processo com suas atividades (MARTÍNEZ-RUIZ et al., 2011).
- BPMN - *Business Process Model and Notation* padrão para modelagem de processos de

negócios e fornece uma notação gráfica para a especificação de processos de negócios em um *Business Process Diagram* (BPD), ou Diagrama de Processos de Negócio, baseado em uma técnica de fluxograma muito semelhante ao de diagramas de atividades da UML.

Destas abordagens de modelagem de processos de software, somente a abordagem de DiNitto e o UML4SPM suportam a capacidade de execução do processo modelado, e com algumas restrições a abordagem de Chous atendia parcialmente a este requisito (BENDRAOU et al., 2010). Em trabalhos mais recentes, novas abordagens tornaram possível a execução do processo para o SPEM (ALEIXO et al., 2011). Para o âmbito do trabalho, foi escolhida a modelagem vSPEM, por possuir suporte a modelagem de variabilidades, elementos integrantes em uma LPrS e, segundo (CARVALHO et al., 2014), por ser a segunda linguagem de modelagem mais utilizada. A primeira seria o *feature model*, também utilizado e disponível no apêndice C.

Ao lado esquerdo da Figura 4 ficam as definições reutilizáveis, chamadas de conteúdo do método, respondendo a questões sobre: “quem” deve fazer “o que” e “como”, sem envolver informações sobre o tempo. Entre os tipos de elementos do conteúdo do método estão: a definição de um produto de trabalho, definição de papéis, definição de tarefas, categorias e guias. Os papéis definem as funções das pessoas envolvidas, as quais podem realizar as tarefas como: Executor principal também chamado de responsável; Adicional também chamado de participante; e Assistente. Cada tarefa possui entradas e saídas, que são relacionadas com os produtos de trabalho, podendo ter guias para detalhar sua elaboração e utilização de ferramentas.



**Figura 4.** Elementos básicos do SPEM, adaptado de (OMG, 2008).

Já ao lado direito da Figura 4 estão os elementos para representar o processo, “quando”

estes são combinados para definir todas as sequências de fases, iterações, o fluxo de atividade e marcos que definem o ciclo de desenvolvimento envolvendo aspectos temporais. Um processo é representado por uma coleção organizada de atividades. Já as atividades são um agrupamento que usam diferentes papéis, produtos de trabalho e tarefas. Os componentes de definição, no conteúdo do processo, podem ser combinados e reutilizados gerando diferentes processos (OMG, 2008).

## **2.2. Linha de Processo de Software**

A abordagem de Linha de Processo de Software (LPrS) surgiu como uma técnica de reuso sistemático de processo e foi apresentada nas abordagens de (BARRETO et al., 2010; JAUFMAN e MÜNCH, 2005; ROMBACH, 2005; WASHIZAKI, 2006) que aplicam a ideia de LPS (Linha de Produto de Software) para processos.

Washizaki (WASHIZAKI, 2006) define uma linha de processo como “uma série de processos para um problema de domínio particular, ou um propósito particular, que eles têm características comuns e são construídas baseados em ativos dos processos reusáveis”. Algumas melhorias podem ser apontadas por meio do uso de LPrS: aumento crescente da produtividade na atividade de definição do processo, diminuindo o esforço para realizá-lo; aumento na qualidade e adequação dos processos gerados pelo reuso do conhecimento de profissionais e uso dos dados; aumento do potencial de reuso de variabilidades por meio da representação; e redução no risco de definição de processos inadequados (JAUFMAN e MÜNCH, 2005; ROMBACH, 2005; WASSERMAN, 2010).

Rombach (ROMBACH, 2005) lida com o conceito de Linha de Processo de Software com base no mesmo princípio de Linha de Produtos de Software. Ele define as principais características de uma Engenharia de Linha de Processos de Software:

1. Dois processos de desenvolvimento separados: o processo de Engenharia de Domínio, em que um (conjunto de) processo (s) genérico (s) para reutilização é projetado para capturar as semelhanças e variabilidade controlada em um domínio, e o processo de Engenharia de Aplicação, por quais processos específicos de um projeto a ser desenvolvido;
2. Um repositório para fornecer processos reutilizáveis de todos os níveis de abstração;
3. Um processo de reuso sistemático, a escolha pré-definida de cada variabilidade, a escolha dos componentes do processo padrão, e;
4. Um processo sistemático de gestão, em que para cada exceção (ex., um comportamento inesperado do processo ocorre) será decidido se a exceção deve ser adicionada ao processo padrão ou não.

### **2.2.1. Engenharia de Domínio (ED)**

Segundo Arango e Prieto-Diaz (ARANGO e PRIETO-DIAZ, 1991), a ED é o processo de identificação e organização do conhecimento sobre uma classe de problemas, o domínio do problema, para suportar sua descrição e solução. O conceito de domínio pode ser entendido como uma classe de sistemas que apresentam funcionalidades similares.

O processo de ED pode ser dividido em três fases principais: Análise do Domínio, Projeto do Domínio e Implementação do Domínio (GRISS et al., 1998; KANG et al., 1990; SIMOS e ANTHONY, 1998). A etapa de Análise do Domínio abrange a identificação, aquisição e representação do conhecimento referente a um domínio, destacando suas variabilidades dentro de uma família de aplicações. A especificação da infraestrutura, ou Projeto do Domínio, gera como resultado uma arquitetura que permite identificar os ativos de um domínio, elementos relacionados ao ciclo de vida de um software que foram projetados para utilização em diferentes contextos (i.e., componentes que são relevantes para a reutilização, a forma de relacionamento entre eles, assim como as formas de customização disponíveis, a distribuição destes componentes pelos repositórios e outras especificações). A Implementação do Domínio compreende o processo de geração dos modelos que podem ser implementados que inclui a identificação, aquisição, extração de sistemas existentes, adaptação e/ou produção e manutenção dos artefatos reutilizáveis.

As atividades envolvidas na ED consistem no processo de desenvolvimento para reutilização que gera artefatos reutilizáveis pelo processo de desenvolvimento com reutilização, caracterizado pela Engenharia de Aplicação (EA). A engenharia de processos da linha de processo de software pode ser observada na Figura 5.

### **2.2.2. Engenharia de Aplicação (EA)**

A EA consiste, inicialmente, na avaliação do domínio e da afinidade da família de sistemas com o produto ou aplicação que será construído. A seguir, a seleção dos componentes necessários à aplicação é feita em um alto nível de abstração, por meio do modelo do domínio, descendo gradualmente em níveis de abstração, até conseguir atingir os componentes implementados ou semidesenvolvidos do domínio (MILER, 2000). Esse recorte é fortemente influenciado pela variabilidade modelada no domínio, uma vez que as características variáveis em artefatos do domínio determinam o tipo de aplicação que será instanciada (OLIVEIRA, 2006).

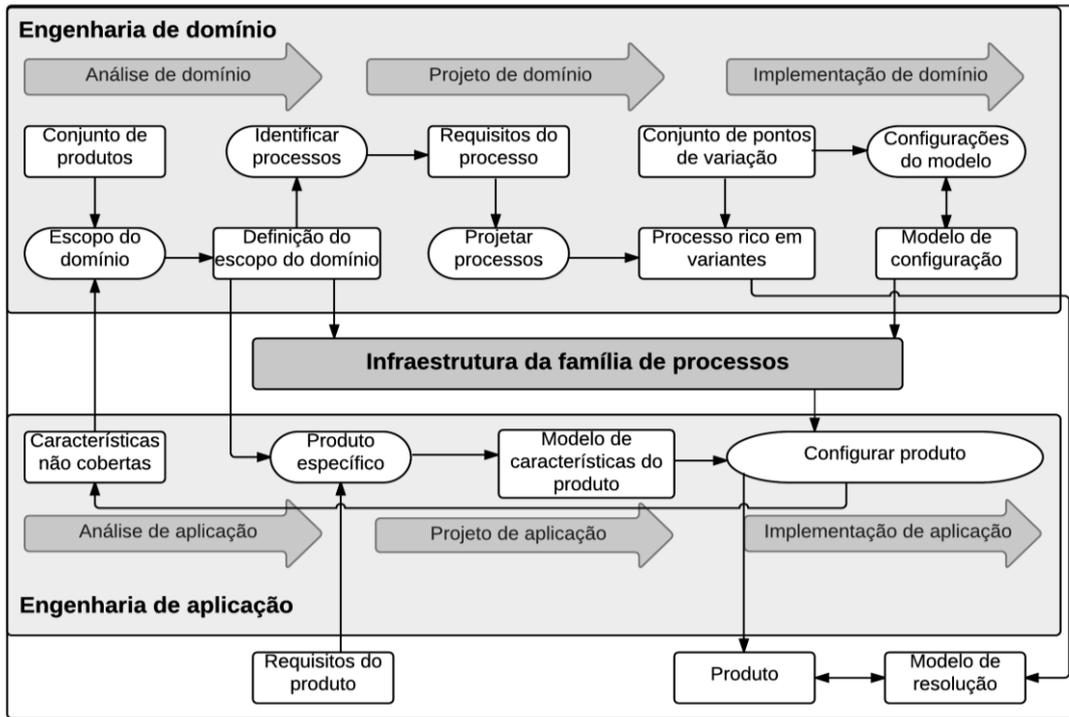


Figura 5. Engenharia de processos para LPrS. Adaptado de (BAYER et al., 2005).

### 2.3. Processo de Teste de Software

A norma internacional ISO/IEC 29119 (ISO/IEEE 29119-2, 2013) disponibiliza um framework geral de processos de teste contendo processos básicos e atividades para garantir a qualidade dos processos gerados (Veja Figura 6).

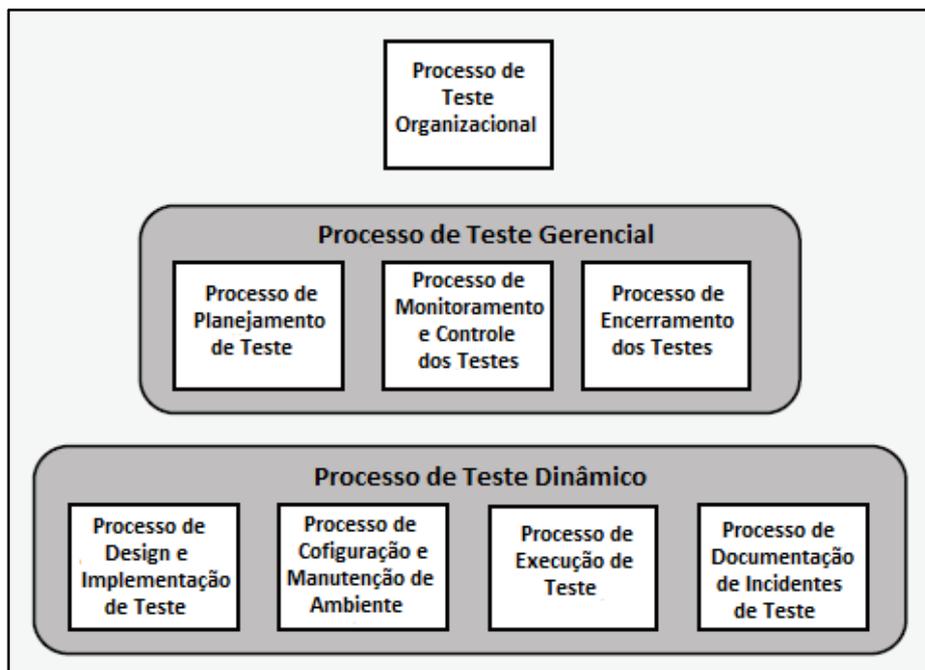
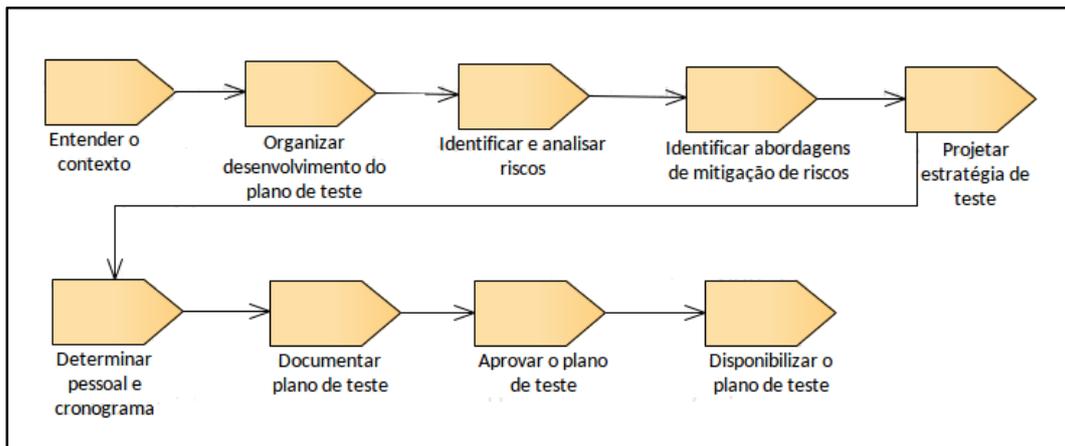


Figura 6. Modelo em camadas do processo de teste. Adaptado de (ISO/IEEE 29119-2, 2013)

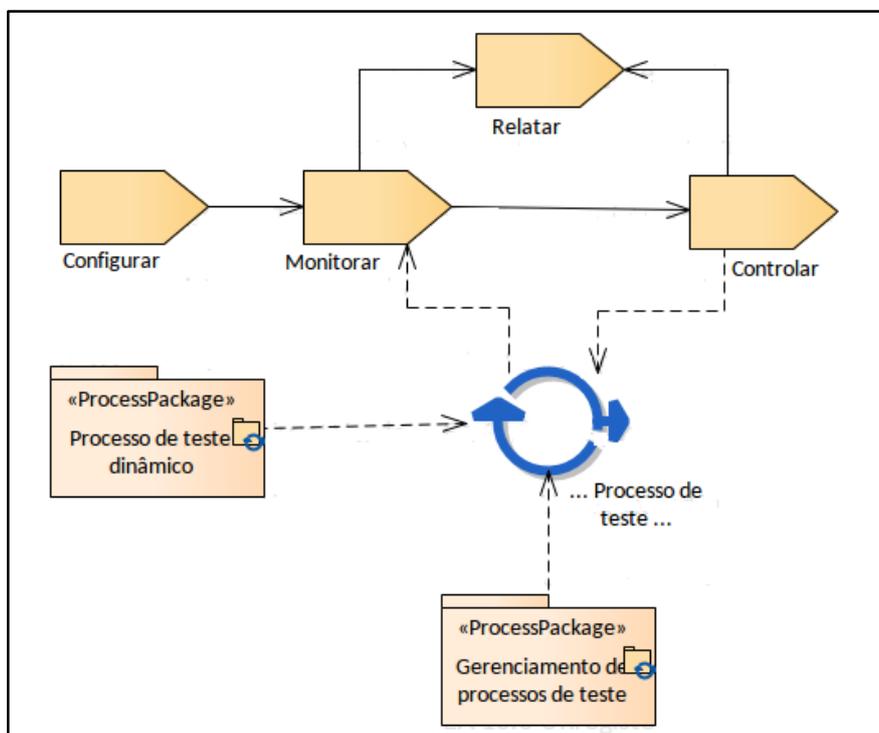
Os processos que compõem a norma são:

- **Processo de Planejamento de Teste (Figura 7):** inclui o entendimento do contexto, organizar desenvolvimento do plano de teste, identificar e analisar riscos, identificar mitigação dos riscos identificados, identificar estratégia de projeto de teste, determinar equipe e cronogramas, documentar o plano de teste, obter aprovação do plano de teste e disponibilizar o plano de testes;



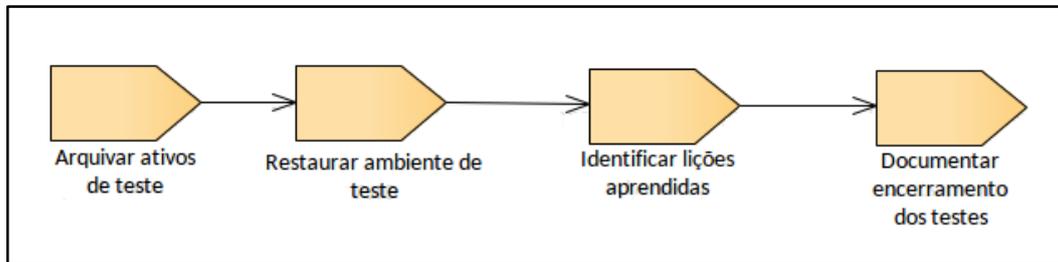
**Figura 7.** Processo de Planejamento de Teste. Adaptado de (ISO/IEEE 29119-2, 2013).

- **Processo de Monitoramento & Controle de Testes (Figura 8):** incluindo configuração, monitoramento, controle e documentação;



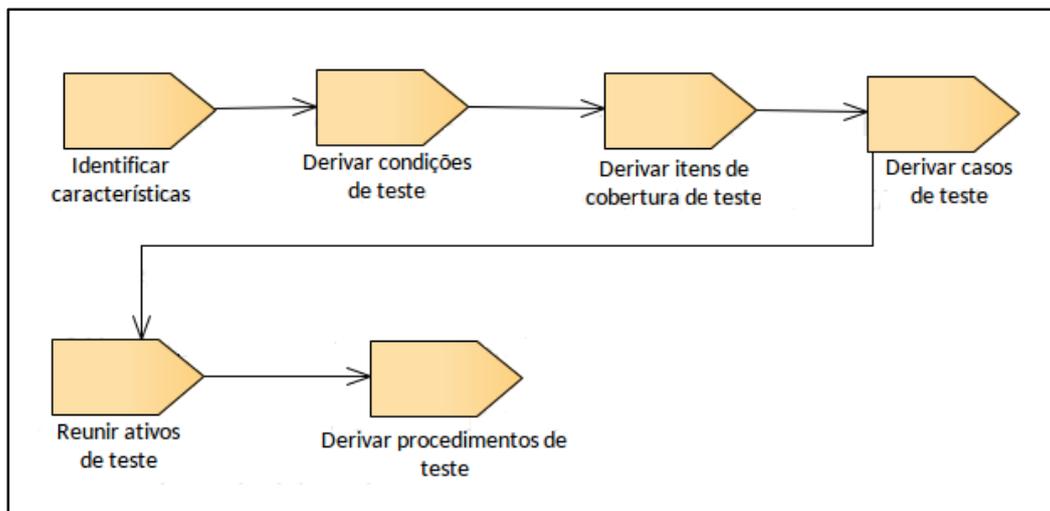
**Figura 8.** Processo Monitoramento e Controle. Adaptado de (ISO/IEEE 29119-2, 2013).

- **Processo de Encerramento de Teste (Figura 9):** incluindo arquivar ativos de teste, restaurar ambiente de testes, identificar lições aprendidas e documentar encerramento dos testes;



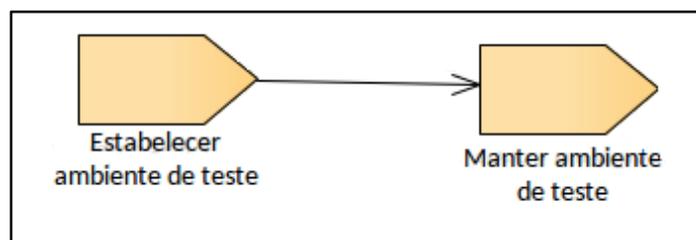
**Figura 9.** Processo de encerramento de testes. Adaptado de (ISO/IEEE 29119-2, 2013).

- **Processo de Projeto & Implementação de Teste (Figura 10):** incluindo identificar conjunto de características, derivar condições de teste, derivar itens de cobertura de testes, derivar casos de teste, reunir conjunto de testes, e derivar procedimentos de teste;



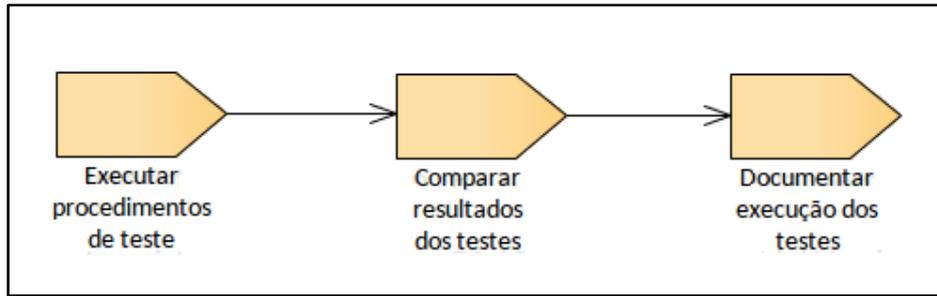
**Figura 10.** Processo de Projeto e Implementação. Adaptado de (ISO/IEEE 29119-2, 2013)

- **Processo de Definição & Manutenção de Ambiente de Teste (Figura 11):** incluindo estabelecimento de ambiente de teste e manutenção;



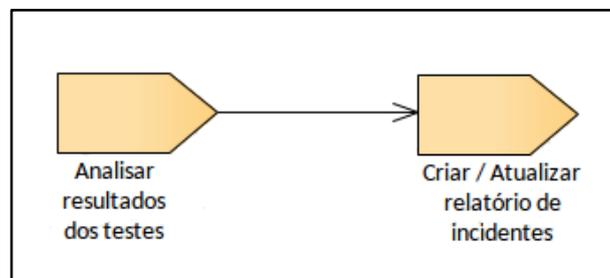
**Figura 11.** Processo de Definição & Manutenção de Ambiente de Teste. Adaptado de (ISO/IEEE 29119-2, 2013).

- **Processo de Execução dos Testes (Figura 12):** incluindo executar procedimentos de teste, comparar resultados dos testes e documentar execução dos testes;



**Figura 12.** Processo de Execução dos Testes. Adaptado de (ISO/IEEE 29119-2, 2013).

- **Processo de Documentação de Incidentes (Figura 13):** incluindo a análise de resultados de testes e criar/atualizar documento de incidentes de testes.



**Figura 13.** Processo de Documentação de Incidentes. Adaptado de (ISO/IEEE 29119-2, 2013).

Apesar de apresentar um framework de processos, a ISO/IEC 29119 (ISO/IEEE 29119-2, 2013) não tem o objetivo de instanciar os processos em uma organização de software. Desse modo, cada organização pode adotar um processo que atenda suas necessidades e especificações. Nesse trabalho é realizada uma investigação das diferentes instanciações do processo de teste de software publicados na literatura técnica por meio de um estudo secundário.

## 2.4. Mapeamento Sistemático sobre Processos de Teste de Software

Na seção 2.3 foram apresentados os conceitos de processos de teste e a ISO/IEEE 29119, porém percebeu-se a necessidade de mapear a literatura técnica existente sobre processos de teste para identificar variações utilizadas na composição dos processos de reuso sistematizado.

Esta seção compartilha resultados de um mapeamento sistemático planejado e executado como parte desta pesquisa, que investiga a literatura existente sobre processos de teste de software, apontando papéis, atividades e testes que o compõem. Inicialmente pretendia-se realizar o mapeamento a respeito de processos de teste para aplicações móveis, porém após alguns testes de *strings* de busca poucos trabalhos foram retornados, inviabilizando assim a restrição a

aplicações móveis. Sendo assim decidiu-se por aumentar o contexto da *string* para processos de teste em geral e após análise identificar quais poderiam ser ajustados para o teste de aplicações móveis.

Um estudo de mapeamento sistemático (SMS - *Systematic Mapping Study*) é planejado e executado como forma de prover uma visão geral da área de pesquisa e determinar uma evidência de pesquisa em tópicos existentes (KITCHENHAM e CHARTERS, 2007). O protocolo definido para o mapeamento sistemático é apresentado na próxima seção.

## 2.4.1. Protocolo do mapeamento sistemático

### 2.4.1.1. Questões de pesquisa

O objetivo deste estudo será esquematizado a partir do paradigma GQM (*Goal Question and Metric*) (BASILI e ROMBACH, 1994), o qual é apresentado na tabela 1. O objetivo deste trabalho é caracterizar o cenário de processos de teste de software, as aplicações, variações, ferramentas, métodos, técnicas, fases e tipos de testes. Para responder ao objetivo desta pesquisa, três relevantes questões de pesquisa (RQs - *Research Questions*) foram definidas:

(RQ1) Quais papéis fazem parte dos processos?

(RQ2) Quais fases compõem o processo?

(RQ3) Quais os tipos de testes aplicados?

**Tabela 1.** Objetivo do mapeamento sistemático segundo o paradigma GQM.

<b>Analisar</b>	O cenário de processos de teste de software
<b>Com o propósito de</b>	Caracterizar
<b>Com relação a</b>	Papéis e atividades que o compõem, tipos de teste e ferramentas utilizadas.
<b>Do ponto de vista do</b>	Pesquisador
<b>No contexto de</b>	Pesquisas acadêmicas e científicas em engenharia de software.

### 2.4.1.2. Identificação e Seleção de Estudo Primários

A *string* de busca utilizada para identificar estudos primários é descrita abaixo:

```
("software testing process" OR "system test process" OR "system testing process" OR "software testing guideline") AND (LIMIT-TO (SUBJAREA,"COMP") OR LIMIT-TO (SUBJAREA,"ENGI"))
```

Foram realizadas buscas nas bibliotecas digitais online: *Scopus*<sup>1</sup>, *IEEEExplore*<sup>2</sup>, *ScienceDirect*<sup>3</sup> e *ACM Digital Library*<sup>4</sup>.

Os critérios de inclusão de estudos para avaliar e/ou analisar os estudos identificados são: O artigo deve estar escrito em Inglês ou Português; Publicações devem estar disponíveis na internet ou poder ser obtidas por meio de contato com seus autores; Publicações devem apresentar e/ou aplicar um processo de teste de software. Qualquer artigo que não atenda todos os critérios de inclusão será excluído.

### 2.4.1.3. Formulação de Extração de Dados

Para cada artigo selecionado foram extraídas informações necessárias para a caracterização de processos de teste de software, essas informações estão descritas na Tabela 2.

**Tabela 2.** Formulário de extração de dados

Título	O título do trabalho.
Autor(es)	O(s) autor(es) do trabalho.
Ano	Ano da publicação do trabalho.
Fonte	Conferência ou revista científica onde o trabalho foi publicado.
Resumo	Resumo introdutório do trabalho.
Papéis	Papéis e responsáveis envolvidos no processo.
Métodos	Métodos utilizados no processo.
Atividades	Atividades que compõem o processo.
Tipo de teste	Tipos de teste aos quais podem ser utilizados no processo.
Inspirado em	Processo utilizado para derivação do trabalho.

### 2.4.2. Execução do Estudo do Mapeamento Sistemático

Todas as atividades do SMS foram realizadas entre maio e setembro de 2015. A Tabela

<sup>1</sup> <http://www.scopus.com>

<sup>2</sup> <http://ieeexplore.ieee.org/Xplore/home.jsp>

<sup>3</sup> <http://www.sciencedirect.com/>

<sup>4</sup> <http://dl.acm.org/>

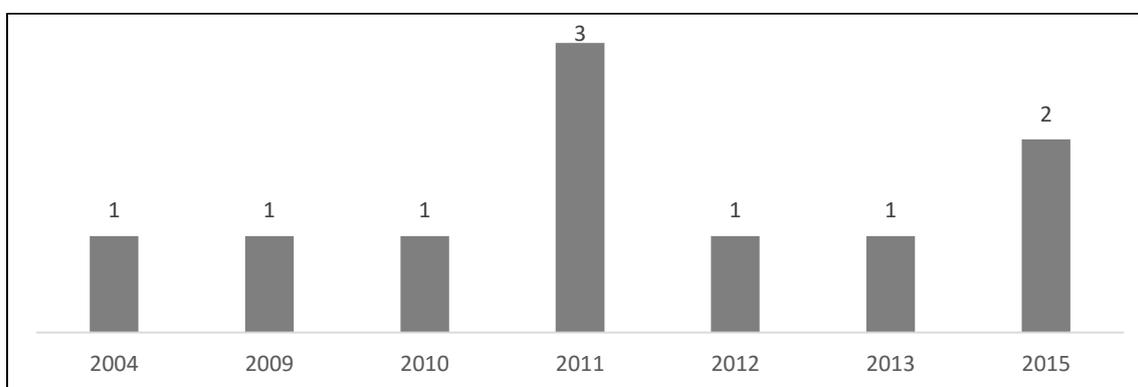
3 apresenta o número de artigos recuperados e selecionados em cada fase do estudo. Foram encontrados 831 artigos nas bibliotecas digitais definidas no protocolo do SMS.

**Tabela 3.** Número de artigos selecionados por fase

Fonte	Artigos recuperados	1º filtro	2º filtro
Scopus	467	86	7
IEEEExplorer	133	10	0
ScienceDirect	147	18	1
ACM Digital Library	84	21	2
<b>Total</b>	<b>831</b>	<b>135</b>	<b>10</b>

Após a aplicação do Filtro 1 (seleção baseada no título, palavras-chave e resumo), 135 artigos foram selecionados. No passo seguinte, Filtro 2 (análise completa do artigo), 10 artigos (publicados entre 2004 e 2015 como mostrado na figura 14) foram selecionados para a extração de informações. Os artigos selecionados estão listados no Apêndice A.

O grande número de artigos rejeitados deve-se que alguns processos foram publicados em mais de uma conferência/*Journal*, nesses casos foi selecionado o artigo com a versão mais completa. Alguns processos apareciam também em mais de uma fonte (artigos duplicados). Nesse caso eles foram classificados de acordo com a ordem de pesquisa, sendo: *Scopus*, *Science Direct*, *IEEEExplore* e *ACM Digital Library*.



**Figura 14.** Publicações por ano

### 2.4.3. Papéis que participam do processo

Os resultados para a questão RQ1 (Quais papéis fazem parte dos processos?) revelam que todos os estudos (100%) consideram o testador um papel importante no processo, o líder de teste aparece em 70% das publicações, gerente de teste com 60% e desenvolvedor com 10%. O analista de teste e gerente de projetos não foram citados nos trabalhos selecionados. Em alguns casos o líder de teste desempenha as mesmas funções do gerente de teste, por essa razão neste trabalho

consideramos o líder de teste o mesmo papel que o gerente de teste, portanto três papéis foram selecionados: testador, líder/gerente de testes e gerente de projetos.

#### **2.4.4. Fases componentes**

Os resultados para a questão RQ2 (Quais fases compõem o processo?) revelam que todos os trabalhos (100%) consideram planejamento, design e execução fases importantes no processo de teste. As fases de análise e encerramento aparecem em 80%, implementação 70%, automação e documentação com 40%. Após análise consideramos que as fases de design e implementação poderiam ser agrupadas, tendo em vista que possuíam atividades semelhantes. A fase de documentação foi considerada paralela a todas as demais, pois cada fase possui uma atividade relacionada a documentação. Analisando cada fase proveniente dos trabalhos em conjunto com a ISO/IEEE 29119-2 (ISO/IEEE 29119-2, 2013) e (CRESPO et al., 2004), foram extraídas as seguintes atividades que compõem o processo de teste:

- Planejamento
  - Caracterizar testes;
  - Definir atividades/tarefas de teste;
  - Definir tipos e técnicas de teste;
  - Identificar e analisar riscos (contingência e mitigação);
  - Projetar estratégias de teste;
  - Documentar plano de teste;
  - Validar plano de teste.
- Design & Implementação
  - Selecionar itens e características de teste;
  - Descrever estratégias para o projeto de testes;
  - Identificar casos de teste e procedimentos;
  - Priorizar casos de teste e procedimentos;
  - Especificar critérios de aprovação/rejeição de testes;
  - Gerar scripts;
  - Documentar projeto de testes.

- Execução
  - Configurar ambiente de testes;
  - Executar procedimentos de testes;
  - Comparar resultados dos testes;
  - Analisar critério de parada;
  - Documentar execução dos testes;
- Análise e Encerramento
  - Gerar relatório de testes;
  - Restaurar ambiente de testes;
  - Identificar lições aprendidas;
  - Documentar análise e encerramento.

#### **2.4.5. Tipos de teste utilizados**

Os resultados para a questão RQ3 (Quais os tipos de testes aplicados?) revelam que analisando os processos, 80% dos estudos consideram o teste funcional importante no processo, e apenas 10% consideram o teste de Qualidade de Serviço (QoS). Outros tipos de teste não foram citados. Analisando os níveis de teste 80% dos estudos consideram o nível de sistema primordial, 70% consideram o nível de integração e 30% nível de unidade. O teste funcional ainda é a principal preocupação dos gerentes de teste, pois tem o objetivo de garantir que as funções esperadas da aplicação estejam funcionando corretamente. Como segunda preocupação temos a QoS relacionada a questões de segurança e desempenho altamente demandadas pelos usuários. Os demais testes dependem em grande parte do tipo de produto que está sendo testados. Para o restante do trabalho consideramos todos os níveis e tipos, incluindo teste estrutural, teste de usabilidade e teste de conectividade.

#### **2.4.6. Considerações do mapeamento**

Esta seção apresentou o estudo do mapeamento sistemático em processos de teste de software. O objetivo foi prover uma visão geral da literatura técnica existente sobre processos de teste de software. Um total de 10 artigos publicados entre 2004 e 2015 foram incluídos no estudo. Os resultados mostraram que o papel de testador um importante componente do processo, seguido pelo gerente de testes. Cobrindo assim a parte gerencial e técnica do processo, foi incluído ainda o papel de projetista de teste.

Foram identificadas também três fases principais do processo: planejamento, design e execução. A partir desses dados algumas atividades foram mescladas originando assim 4 fases principais: planejamento, design & implementação, execução e análise & encerramento, a serem utilizadas como base posterior do trabalho. Foram integradas a cada fase, atividades provenientes da ISSO/IEEE 29119-2 (ISO/IEEE 29119-2, 2013) e (CRESPO et al., 2004).

Quanto aos tipos de teste mencionados nos processos de teste os principais são: funcionais e QoS, considera-se QoS como testes de segurança e desempenho, foram adicionados à lista de tipos de teste os seguintes: usabilidade, conectividade e estrutural, tendo em vista que o objetivo é uma LPrS voltada ao teste de aplicações móveis e segundo (DELAMARO et al., 2016) esses são considerados tipos de teste essenciais para garantir a qualidade de uma aplicação móvel.

## **2.5. Extensão de mapeamento sistemático sobre Linhas de Processo de Software**

Foi identificada a necessidade de uma pesquisa na literatura a respeito de Linhas de Processo de Software com o objetivo de identificar os principais elementos que sofrem variação, onde a partir dos elementos identificados seriam extraídas variantes relacionadas aos elementos. No entanto, ao iniciar as pesquisas na literatura técnica foi encontrado um mapeamento sistemático abrangendo linhas de processo de software. Sendo assim, realizar um novo mapeamento não seria mais necessário. Porém, o referido mapeamento abrangeu os anos de 1996 até 2013. Assim, observando o crescimento de maturação na área foi realizada uma extensão do mapeamento cobrindo os anos de 2013 a 2017. As mesmas *strings* de busca foram utilizadas e as mesmas bibliotecas digitais.

### **2.5.1. Método de pesquisa**

Para a extensão foi utilizando o mesmo método de pesquisa do mapeamento original, a Revisão Sistemática da Literatura (SLR – *Systematic Literature Review*), considerado um método bem definido com o objetivo de analisar, identificar, avaliar e interpretar todas as evidências disponíveis relacionadas a uma questão de pesquisa específica. O processo de execução da SLR segue as seguintes fases: (i) planejamento, (ii) condução da SLR e (iii) resultados.

#### **2.5.1.1. Planejamento**

A fase de planejamento consiste das seguintes atividades: (i) identificação da necessidade da SLR; (ii) especificação de uma questão de pesquisa; e (iii) desenvolvimento de um protocolo de pesquisa.

**Necessidade de uma extensão do mapeamento sistemático:** Baseando-se que o mapeamento sistemático seguiu-se até o ano de 2013 utiliza-lo seria deixar uma lacuna de anos de pesquisa e maturação da área muito grande, por esse motivo uma extensão do mapeamento foi necessária buscando novas técnicas de gerenciamento de variabilidade, pontos de variação e ferramentas de apoio.

**Questões de pesquisa:** Como trata-se de uma extensão as mesmas questões de pesquisa foram utilizadas. Disponíveis na tabela 4.

**Tabela 4.** Questões de pesquisa

ID	Questão de pesquisa
Q1	Quais as maiores continuações das abordagens propostas?
Q2	Como representar variabilidades?
Q2.1	Quais tipos de variabilidades são fornecidas?
Q2.2	Quais tipos de dependências são fornecidas?
Q2.3	Quais elementos do processo de software são alvos de variabilidades?
Q3	Quais linguagens de modelagem de processo são utilizadas?
Q4	Quais padrões e modelos de qualidade de processo são utilizados?
Q5	Quais paradigmas de engenharia de software são aplicados em conjunto com LPrS?
Q6	Quais conceitos de linha de processo de software são utilizados?
Q7	Existem ferramentas para apoiar essas abordagens?
Q8	Como essas abordagens propostas foram validadas?

**Protocolo da SLR:** O protocolo da SLR documenta o método que será utilizado para conduzir o mapeamento sistemático, como forma de reduzir o viés do pesquisador e possibilitar a reprodução.

**Estratégia de pesquisa:** Foi utilizada a mesma estratégia de pesquisa do mapeamento original, dessa forma as *strings* foram construídas considerando as formas singulares e plurais e possíveis sinônimos dos termos principais.

**Critérios de Inclusão:** Os critérios de inclusão são apresentados na Tabela 5.

**Tabela 5.** Critérios de inclusão

ID	Critérios de inclusão
IC-01	O artigo apresenta variabilidade, similaridade ou flexibilidade em modelos de processos.
IC-02	O artigo apresenta linhas de produto.
IC-03	O artigo apresenta reuso de processos ou adaptação de processos.

<b>IC-04</b>	O artigo apresenta linhas de processo.
--------------	--

**Critérios de Exclusão:** Se um artigo for classificado em ao menos um critérios de exclusão, então ele será rejeitado. A lista de critérios de exclusão é mostrado na Tabela 6.

**Tabela 6.** Critérios de exclusão

<b>ID</b>	<b>Critérios de exclusão</b>
<b>EC-01</b>	O artigo se refere a um relatório técnico, tutorial, minicurso ou algum tipo de trabalho não científico.
<b>EC-02</b>	O artigo não está relacionado à área de Engenharia de Software (ou até mesmo a área de Computação)
<b>EC-03</b>	O artigo apresenta reuso apenas em software.
<b>EC-04</b>	O artigo apresenta variabilidade/similaridade apenas em software ou outra entidade.
<b>EC-05</b>	O artigo emprega variabilidade/similaridade apenas no contexto de comparação genérica, relacionamento de similaridade estrutural, mineração de processo, controle de processo estatístico, variação de processo ou imprevisibilidade de processo.
<b>EC-06</b>	O artigo utiliza o termo “ <i>feature</i> ” apenas no conceito de funcionalidade.
<b>EC-07</b>	O artigo apresenta processos de software, mas não considera a modelagem de variabilidade.
<b>EC-08</b>	O artigo não foi encontrado sem custo.
<b>EC-09</b>	O artigo não foi encontrado.
<b>EC-10</b>	O artigo apresenta o reuso de processo de software, mas sem utilizar os conceitos de LPrS.
<b>EC-11</b>	O artigo não está escrito em inglês.
<b>EC-12</b>	O artigo apresenta apenas linhas de produto.
<b>EC-13</b>	O artigo representa um mapeamento sistemático.

### 2.5.1.2. Execução

Essa fase consiste na execução do protocolo especificado. A fase é composta dos seguintes passos: (i) identificação da pesquisa; (ii) seleção de estudos primários; (iii) extração de dados; e (iv) análise dos dados.

**Identificação da pesquisa:** Nesse passo foram utilizadas as mesmas bibliotecas digitais do estudo original, são elas: *Scopus*, *IEEEExplore* e *Engineering Village*, todas as buscas foram realizadas nos meses de Fevereiro e Março de 2017. As tabelas 7 e 8 apresentam as *strings* utilizadas em cada biblioteca.

**Tabela 7.** *Strings* de busca

**String utilizada na Scopus:** TITLE-ABS-KEY ( ( "process line" OR "process lines" OR "process family" OR "process families" OR "variant rich process" OR "variant-rich process" ) OR ( ("business process" OR "business processes" OR "software process" OR "software processes" OR "process model" OR "process models" ) AND ( variability OR variabilities OR variation OR variations OR variant OR variants OR commonality OR commonalities OR similarity OR similarities OR "feature model" OR "feature models" OR "feature modeling" OR "feature-based" ) ) ) AND ( PUBYEAR > 2013 ) AND ( PUBYEAR < 2018 ) AND ( LIMIT-TO ( SUBJAREA , "COMP " ) )

**String utilizada na Engineering Village:** (((("process line" OR "process lines" OR "process family" OR "process families" OR "variant rich process" OR "variant-rich process") OR (("business process" OR "business processes" OR "software process" OR "software processes" OR "process model" OR "process models") AND (variability OR variabilities OR variation OR variations OR variant OR variants OR commonality OR commonalities OR similarity OR similarities OR "feature model" OR "feature models" OR "feature modeling" OR "feature-based")))) WN KY) AND (723\* WN CL) AND (({ca} OR {ja} OR {ip}) WN DT))

A biblioteca IEEE Xplore apresenta uma limitação no número de termos da pesquisa, sendo assim sua *string* foi dividida em 6 partes e realizado o merge com os resultados, a *string* original e as particionadas são apresentadas na tabela 8.

**Tabela 8.** *Strings* da IEEE Xplore

**String original da IEEE Xplore:** ("process line" OR "process lines" OR "process family" OR "process families" OR "variant rich process" OR "variant-rich process") OR (("business process" OR "business processes" OR "software process" OR "software processes" OR "process model" OR "process models") AND (variability OR variabilities OR variation OR variations OR variant OR variants OR commonality OR commonalities OR similarity OR similarities OR "feature model" OR "feature models" OR "feature modeling" OR "feature-based"))

**Additional Settings:** “Advanced search” and “command search” modes selected. Option “metadata only” checked as true. “Publication year” in mode “range” from 1996 to 2013. “Content Type” with values “Conference Publications”, “Journals & Magazines” and “Early Access Articles” selected.

**Tabela 8** (Continuação)

<p><b>String 1:</b> (("process line" OR "process lines" OR "process family" OR "process families" OR "variant rich process" OR "variant-rich process") OR (("business process" OR "business processes" OR "software process" OR "software processes" OR "process model" OR "process models") AND variabilit* OR commonalit*))</p>
<p><b>String 2:</b> ("process line" OR "process lines" OR "process family" OR "process families" OR "variant rich process" OR "variant-rich process") OR (("business process" OR "business processes" OR "software process" OR "software processes" OR "process model" OR "process models") AND similarit*))</p>
<p><b>String 3:</b> (("process line" OR "process lines" OR "process family" OR "process families" OR "variant rich process" OR "variant-rich process") OR (("business process" OR "business processes" OR "software process" OR "software processes" OR "process model" OR "process models") AND ("feature model")))</p>
<p><b>String 4:</b> (("process line" OR "process lines" OR "process family" OR "process families" OR "variant rich process" OR "variant-rich process") OR (("business process" OR "business processes" OR "software process" OR "software processes" OR "process model" OR "process models") AND ("feature models")))</p>
<p><b>String 5:</b> ("process line" OR "process lines" OR "process family" OR "process families" OR "variant rich process" OR "variant-rich process") OR (("business process" OR "business processes" OR "software process" OR "software processes" OR "process model" OR "process models") AND ("feature-based"))</p>
<p><b>String 6:</b> ("process line" OR "process lines" OR "process family" OR "process families" OR "variant rich process" OR "variant-rich process") OR (("business process" OR "business processes" OR "software process" OR "software processes" OR "process model" OR "process models") AND ("feature modeling"))</p>

**Seleção de estudos primários:** No primeiro filtro apenas título, resumo e palavras-chaves dos artigos foram lidos. Após esse passo 546 artigos foram identificados como duplicados, 898 foram rejeitados e 149 foram aceitos. O grande número de duplicados deu-se pelo particionamento da *string* da biblioteca IEEE Xplore.

No segundo filtro uma análise mais detalhada foi conduzida nos trabalhos aceitos no primeiro filtro, sendo assim foi realizada a leitura completa dos artigos. Após o segundo filtro 121 artigos foram rejeitados, 5 considerados duplicados e 23 foram aceitos.

## 2.5.2. Extração dos dados e resultados

Nesse passo, as informações dos papéis aceitos foram extraídas de forma a responder as questões do mapeamento.

### Contribuições principais (Q1)

Linha de processo modelada é a contribuição principal mais comum (Figura 15). Contribuições como hipóteses, diretrizes e requisitos de variabilidade não foram abordados nos últimos anos em comparação ao estudo original.

Observando ambos os gráficos pode-se perceber que atualmente cresceu a preocupação com a modelagem da LPrS e métodos utilizados, em comparação com o estudo original onde observa-se uma maior preocupação com métodos e notação gráfica. Isto deve-se devido ao fato de que algumas notações gráficas e métodos já encontram-se amadurecidas na literatura, não sendo mais foco de estudo.

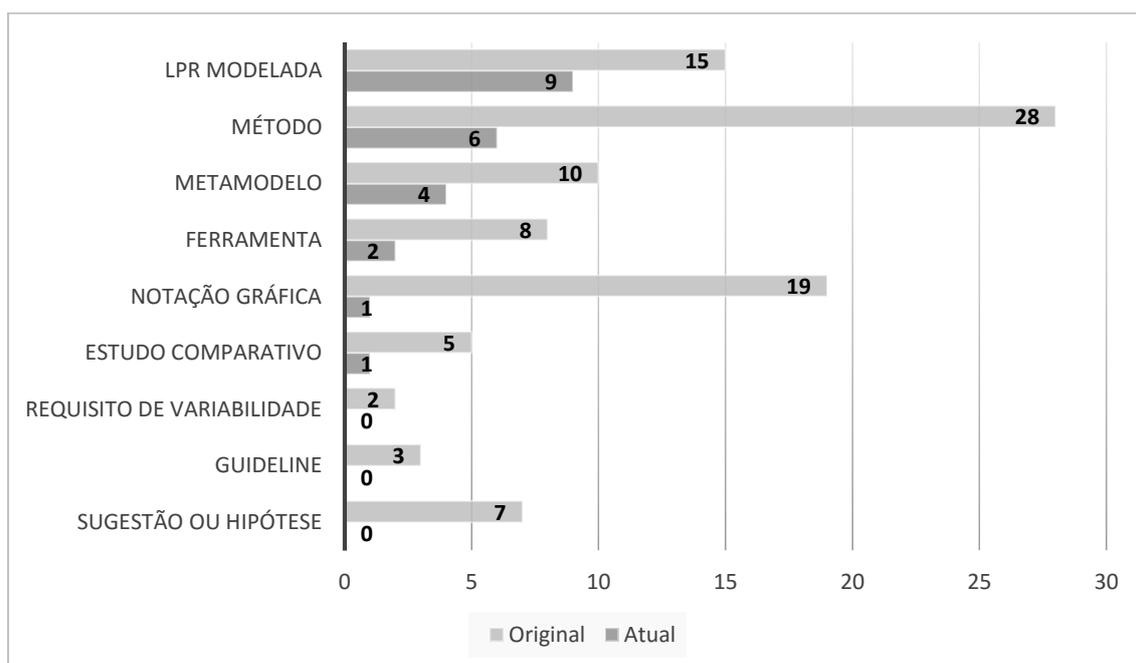
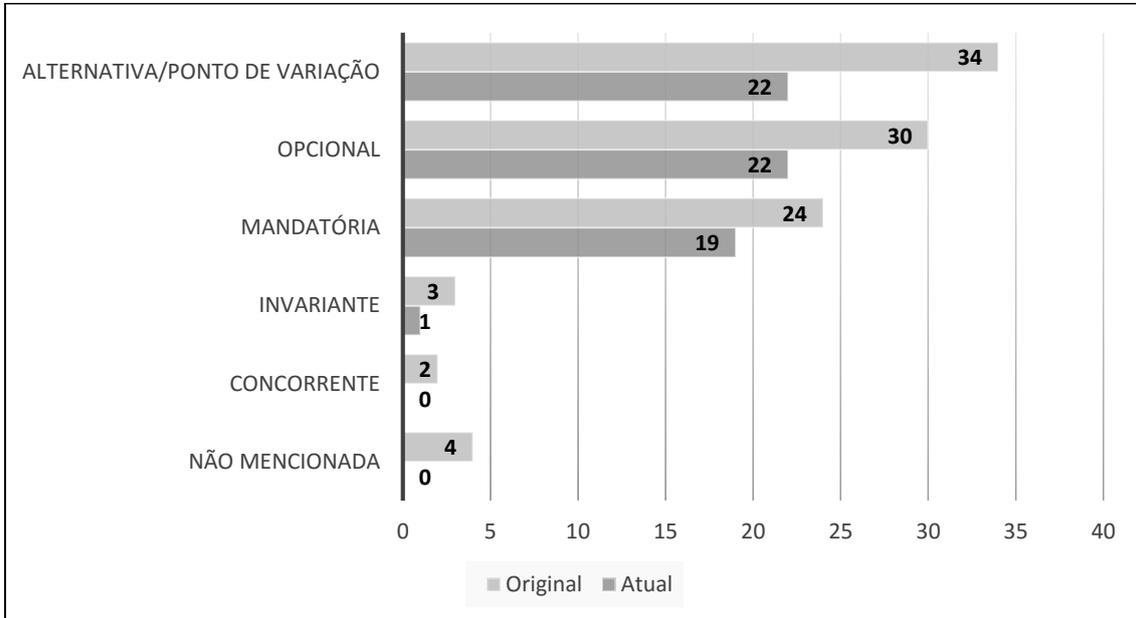


Figura 15. Principais contribuições identificadas.

### Tipos de Variabilidade (Q2.1)

O propósito dessa questão é investigar como as variabilidades são representadas em comparação ao estudo original. Foram consideradas os mesmos tipos de variabilidade identificadas no estudo original, são elas: “alternativa/ponto de variação”, “opcional”, “mandatória”, “invariante” e “concorrentes”. A Figura 16 mostra os tipos de variabilidade encontradas em ambos os estudos.

Ambos os gráficos apresentam informações semelhantes, confirmando que os mesmos tipos de variabilidade continuam sendo as mais utilizadas (alternativa/ponto de variação e opcional), pode-se observar também uma maior preocupação em mencionar os tipos de variabilidade utilizadas.

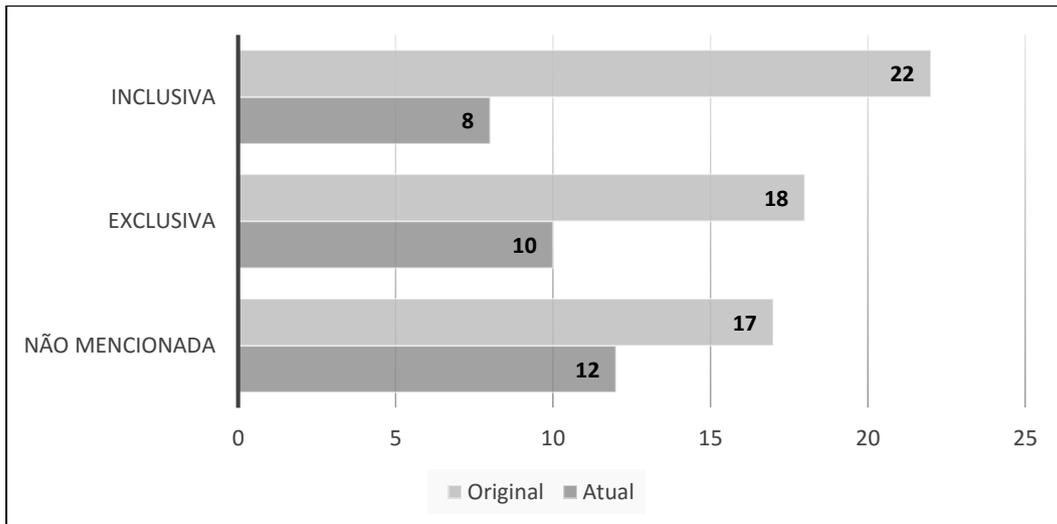


**Figura 16.** Tipos de variabilidade encontradas.

### **Tipos de dependências (Q2.2)**

As dependências são mecanismos para apoiar as decisões nas variabilidades. Foram considerados dois tipos de dependências “inclusiva” (se A é incluída, então B também deve ser) e “exclusiva/exclusão mutua” (se A é incluída, então B não deve ser). A Figura 17 mostra os tipos de dependências do estudo atual e do estudo original.

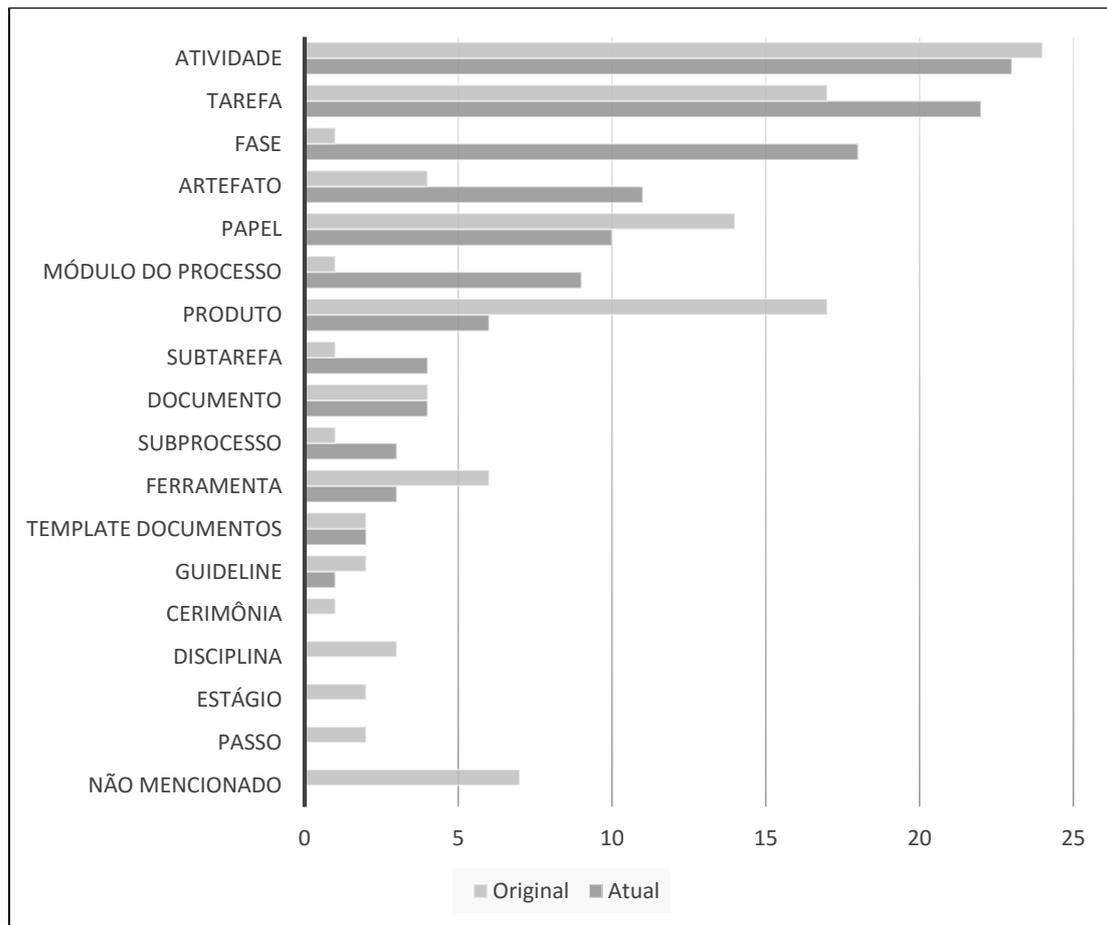
Pode-se então perceber que os tipos de dependências passaram a ser mais citadas nos trabalhos atuais, apresentando assim uma ordem de prioridade inversa da vista no estudo original, onde a mais citada era a dependência de inclusão, aparecendo como menos citada no cenário atual.



**Figura 17.** Tipos de dependências encontradas

### **Elementos do Processo de Software (Q2.3)**

Foram identificados os elementos do processo de teste passíveis de variação apontados pelos estudos, devido ao grande número de elementos de um processo de software não foram pré-determinadas nenhuma opção de resposta. Os resultados dessa questão de pesquisa podem ser observados na Figura 18.

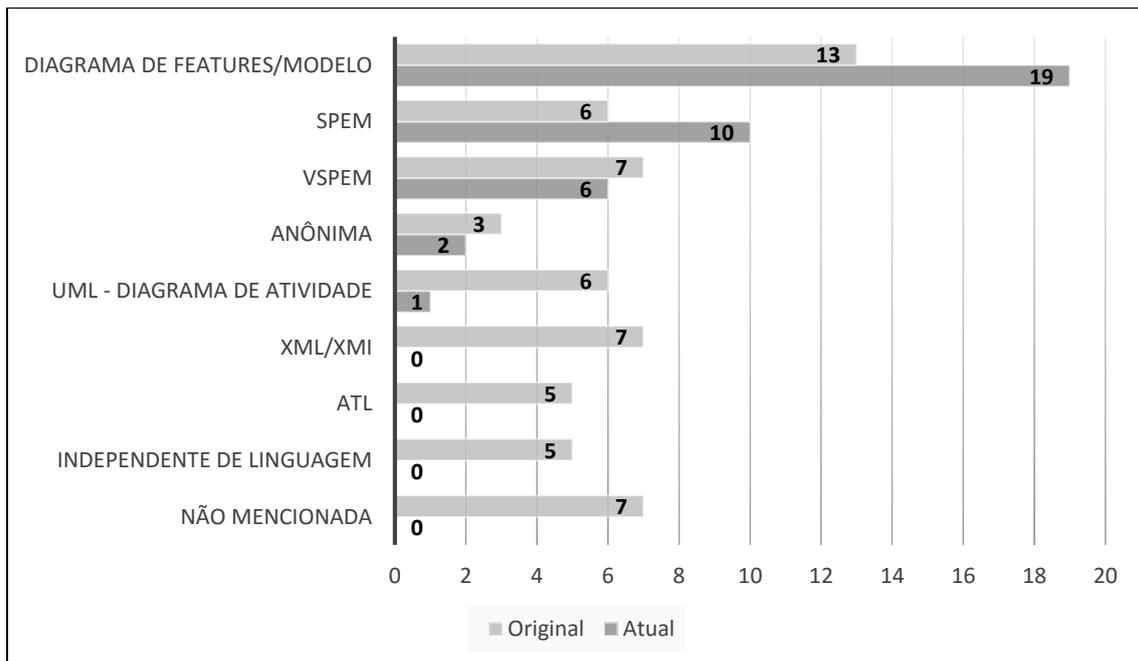


**Figura 18.** Elementos do processo

Os dois principais elementos do processo relevantes a aplicação de variabilidade seguiram a tendência identificada no estudo original. Observa-se também um grande crescimento do elementos fase e um decréscimo na ausência de elementos do processo. Pode-se perceber então que com o passar dos anos há uma maior preocupação com a documentação das linhas de processo de software. O que mostra um amadurecimento exponencial da área quando considerado o espaço do tempo avaliado no estudo atual.

### **Linguagem de modelagem (Q3)**

O objetivo da questão é identificar quais as linguagens de modelagem mais utilizadas na definição de uma linha de processo de software, foram classificadas as oito principais buscando avaliar a evolução das linguagens de modelagem. A quantidade de trabalhos que referenciavam alguma linguagem de modelagem no estudo atual e estudo original são apresentadas na Figura 19.



**Figura 19.** Linguagens de modelagem identificadas.

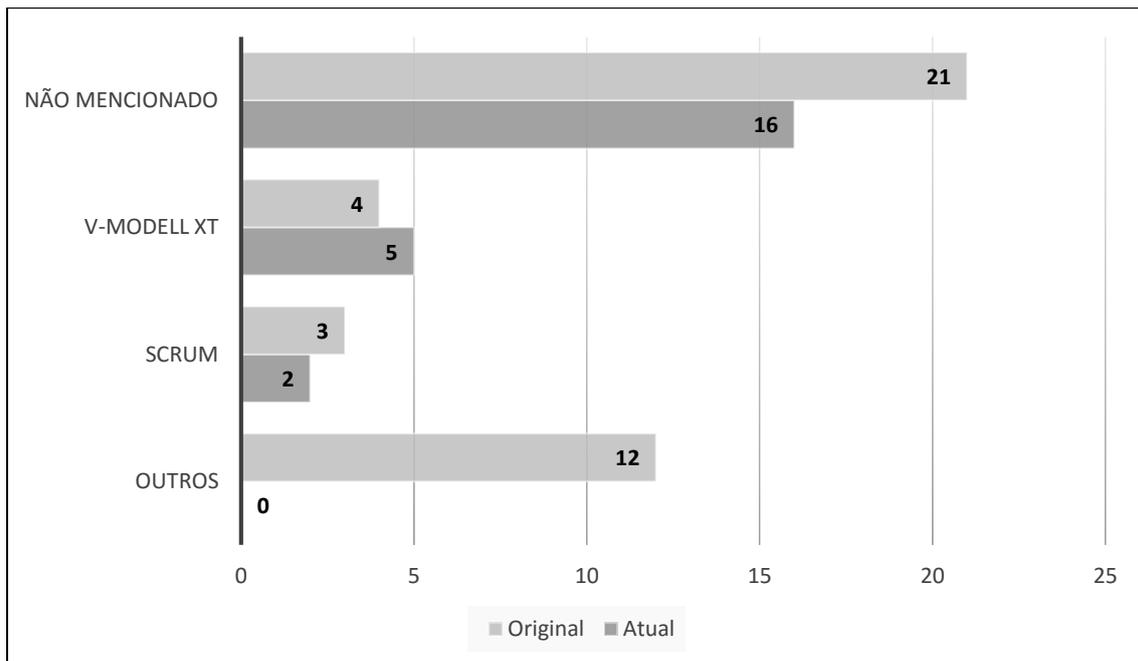
O diagrama de *features* continua sendo a principal linguagem de modelagem de linhas de processo de software, seguido pelo SPEM, observou-se uma queda na utilização da vSPEM e o diagrama de atividades UML. Percebeu-se que cada vez mais os trabalhos se preocupam em especificar uma linguagem de modelagem para apresentar uma modelagem visual da linha de processo de software.

#### **Modelos e padrões de qualidade de processo (Q4)**

Quando uma organização deseja certificar ou avaliar seu processo de desenvolvimento de software, o processo deve estar definido rigorosamente e em conformidade com os padrões e modelos mais populares (ex.: CMMI-DEV, MR-MPS-SW, ISSO/IEC 12207) e a LPrS é uma abordagem suscetível a isso.

A extensão do mapeamento identificou 2 modelos de qualidade de processo e padrões citados, percebendo-se assim uma menor preocupação em avaliar a qualidade dos processos gerados pelas linhas apresentadas quando comparado ao estudo original onde foram citados 19 modelos. Os resultados dessa questão podem ser observados na Figura 20.

É importante salientar que no estudo original muitos padrões e modelos de qualidade eram aplicados apenas a exemplos, não sendo utilizados na avaliação da linha de processo resultante do estudo. Isso pode ser utilizado como evidencia para explicar o porquê da diferença tão grande da aplicação ou não de modelos de qualidade.

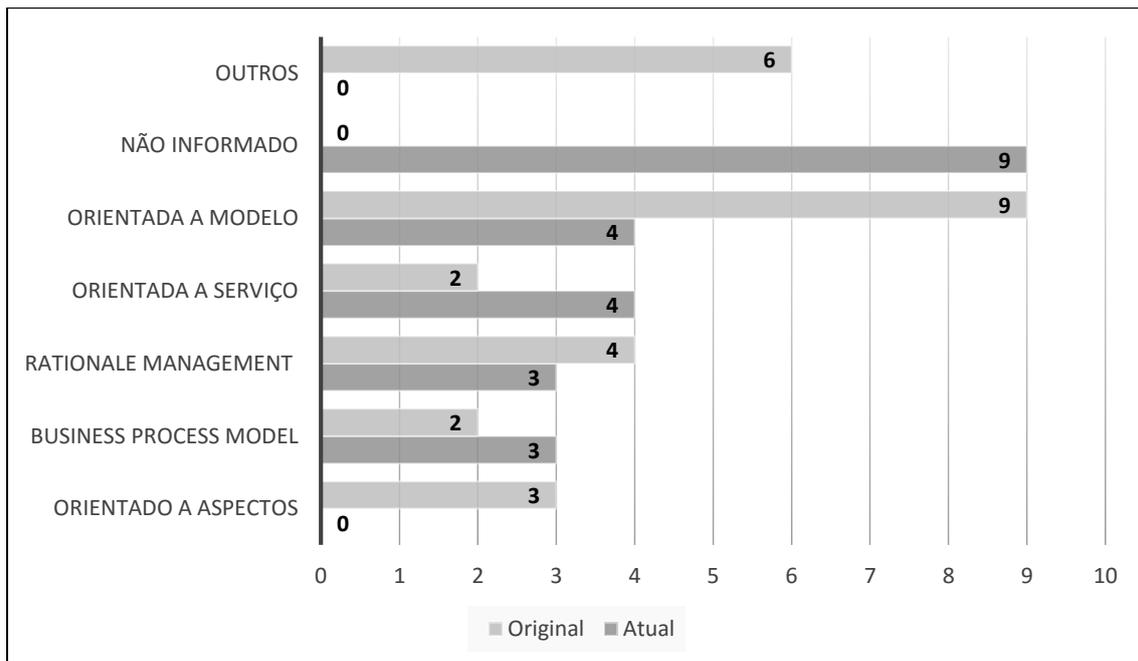


**Figura 20.** Modelos e padrões de qualidade de processo identificados

### **Paradigmas de engenharia de software (Q5)**

Há alguns paradigmas de engenharia de software que podem ser utilizados para auxiliar a elaboração de uma linha de processo de software. Normalmente são aplicados em conjunto a LPrS buscando gerar maior qualidade nos resultados e até mesmo na definição e identificação de requisitos da linha. Alguns paradigmas como *Model-Driven Engineering* (MDE) auxiliam na automatização de alguns passos da engenharia de LPrS. Dessa forma a questão tem como objetivo identificar os paradigmas aplicados atualmente em comparação com os utilizados no estudo original, avaliando se novos paradigmas estão sendo aplicados. Os resultados da questão podem ser observados na Figura 21.

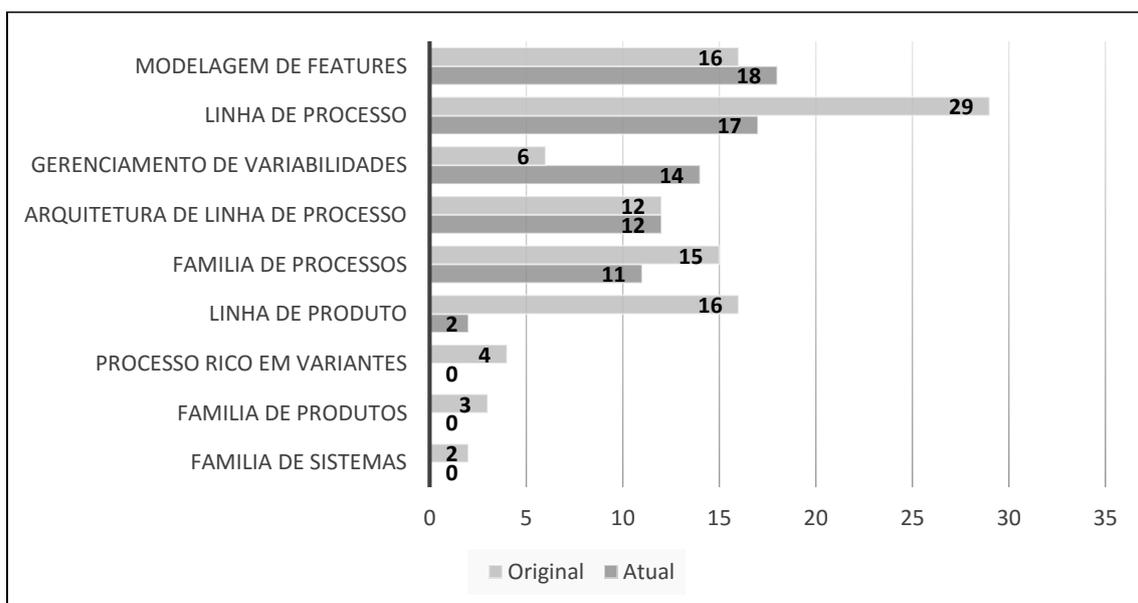
Analisando a Figura 21, pode-se perceber que muitos trabalhos deixaram de se preocupar com a adoção de paradigmas de engenharia de software, percebe-se ainda um crescimento na adoção do *Business Process Model* (BPM) e Orientada a serviço e uma diminuição na adoção de orientação a aspectos e *Rationale Management*.



**Figura 21.** Paradigmas de engenharia de software

### Conceitos de Linhas de Processo de Software utilizados (Q6)

Uma linha de processo de software possui uma série de conceitos principais que podem ser utilizados dependendo do objetivo da linha proposta. Foram selecionados 9 conceitos principais passíveis a respostas desta questão, após coleta foi realizada análise comparativa entre os resultados no estudo atual e no estudo original ao qual foi proposta a extensão. Os resultados podem ser analisados na Figura 22.

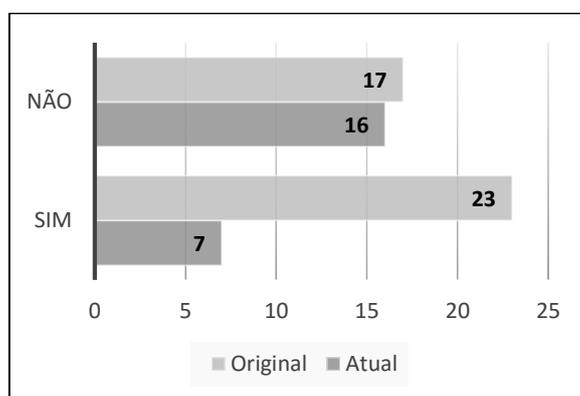


**Figura 22.** Conceitos de linha de processo utilizados

Comparando o estudo original com o anterior percebe-se uma maior adoção aos conceitos de modelagem de *features* componentes da linha, um dos fatores que alavanca essa maior adoção deve-se a maior preocupação na representação dos requisitos da linha em busca de maior qualidade. Há também um maior conjunto de trabalhos que adotam o gerenciamento de variabilidades como um dos conceitos principais a serem aplicados, por outro lado conceitos como linha de produto e família de sistema vem sendo menos utilizados.

### Ferramentas de suporte (Q7)

Ferramentas existentes na literatura podem ser utilizadas na elaboração ou validação de LPrS, com o crescimento do conceito de automatização espera-se que cada vez mais sejam utilizadas mais ferramentas de suporte a LPrS. Os resultados da questão podem ser observados na Figura 23.



**Figura 23.** Ferramentas de suporte

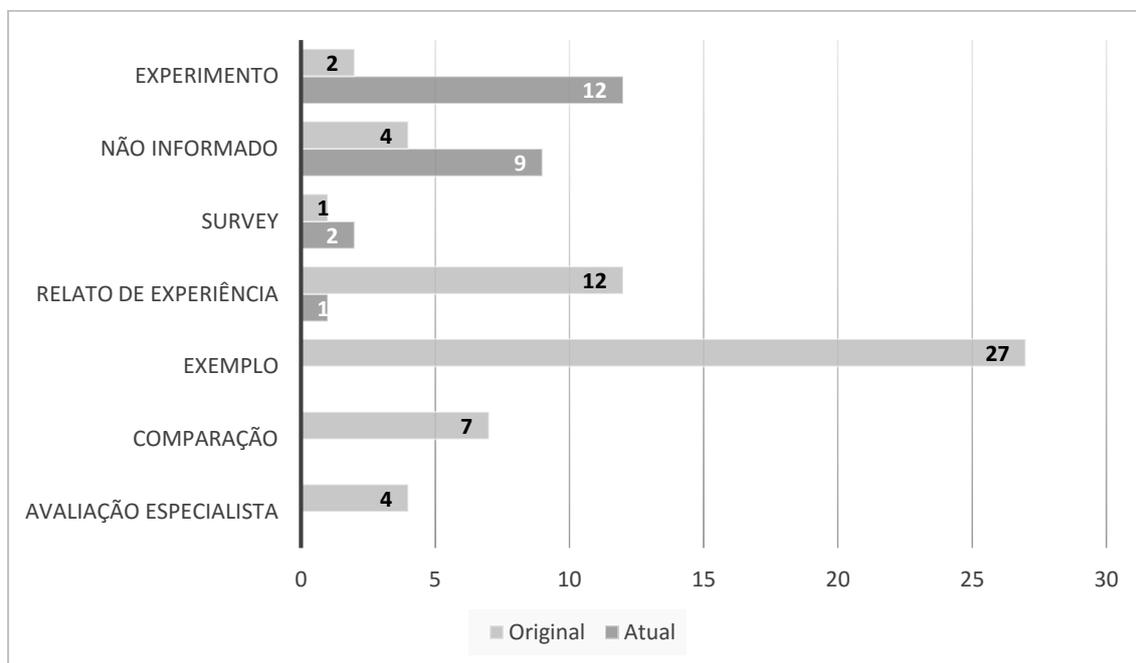
Nos resultados atuais poucos trabalhos utilizavam ferramentas de suporte em comparação ao estudo original, isso deve-se ao fato da maior preocupação atual com as fases iniciais de implementação de um LPrS, modelagem por exemplo, buscando assim uma diminuição na quantidade de retrabalho necessário para sua elaboração.

### Métodos de validação (Q8)

Uma das principais questões na área de LPrS atualmente é como realizar a validação da linha implementada, várias técnicas e métodos vem sendo aplicados no decorrer do processo de maturação da área. A adoção do método adequado é um ponto de grande discussão, tendo em vista que muitos deles necessitam de um longo tempo de aplicação e observação, dessa forma os métodos acabam sendo aplicados de acordo com o tempo e recursos disponíveis para validação. Foram identificados 6 métodos principais de validação que podem ser observados na Figura 24.

Analisando a Figura 24, pode-se perceber o abandono do exemplo como método de

validação e a adoção de experimento como principal meio, porém poucos trabalhos apresentam um método de validação para as LPrS elaboradas. Afirmando a atual pouca ênfase as fases finais de elaboração de uma linha de processo de software.



**Figura 24.** Métodos de validação utilizados

Dessa forma o mapeamento sistemático realizado tinha o objetivo de complementar e analisar o estado atual da área de LPrS, baseando-se em um corpo de conhecimento atualizado e consistente. A importância do mesmo é percebida ao realizar a comparação dos resultados dos dois estudos, enfatizando a mudança de prioridades de acordo com a fase de maturação da área.

## 2.6. Teste de Aplicações Móveis

Uma aplicação móvel é um software projetado para executar em *smartphones*, *tablets* e outros dispositivos móveis e/ou considerando informações contextuais de entrada (CHEN e KOTZ, 2000).

De modo geral, as aplicações móveis podem ser classificadas em três categorias (MASI et al., 2012): aplicações nativas, aplicações web ou aplicações híbridas. Aplicações nativas são aquelas especificamente concebidas para serem executadas em um sistema operacional e *firmware* do dispositivo e, normalmente, têm de ser adaptadas para diferentes dispositivos. Elas possuem acesso a recursos internos do dispositivo (ex.: câmera, GPS, acelerômetro etc.). Uma aplicação web é aquela em que todas ou algumas partes do software são baixadas da Internet cada vez que a aplicação é executada a partir do conteúdo disponibilizado em um servidor web. Geralmente, ela pode ser acessada a partir de todos os dispositivos móveis que possuem acesso à

Internet por meio de um navegador web (browser) e, em geral, não possui acesso aos recursos internos do dispositivo móvel. Por fim, uma aplicação híbrida (HTML5 e aplicações *widgets*) possui características de ambas as categorias anteriormente descritas. Como uma aplicação nativa, ela também deve ser instalada e possui acesso a recursos internos do dispositivo móvel; por outro lado, como uma aplicação web, ela pode ser baseada na linguagem HTML5 e ser exibida em um navegador web, tendo todas ou algumas partes do seu conteúdo baixadas da web. Aplicações híbridas permitem o desenvolvimento multiplataforma usando o mesmo código HTML5 para diferentes sistemas operacionais. Elas ainda permitem a compilação do código em formato nativo, reduzindo, assim, os custos de desenvolvimento.

Além dos aspectos presentes em qualquer software (ex.: funcionalidade), diversos aspectos específicos de uma aplicação móvel são passíveis de teste, como o seu comportamento, a interação dos usuários com a aplicação, aspectos de conectividade e mobilidade, visto que essas são características fundamentais em uma aplicação móvel. Assim, diferentes abordagens de teste são propostas para esse conjunto de aspectos que torna uma aplicação móvel diferente de outras categorias de software.

Teste de software para aplicações móveis se refere aos diferentes tipos de teste a serem aplicados para os diferentes tipos de aplicações (nativa, híbrida e web) que executam sobre a plataforma móvel usando métodos e ferramentas de teste de software bem definidos a fim de garantir a qualidade em funções, comportamentos, desempenho e qualidade de serviço, assim como atributos como: mobilidade, usabilidade, interoperabilidade, conectividade, segurança e privacidade (GAO et al., 2014).

Em muitos aspectos, o desenvolvimento de aplicações móveis é semelhante à Engenharia de Software para outras plataformas. Problemas comuns incluem integração com o hardware do dispositivo, bem como as questões tradicionais de segurança, desempenho, confiabilidade e limitações de armazenamento. No entanto, as aplicações móveis apresentam alguns requisitos adicionais que são menos comumente encontrados em aplicações de software tradicionais. Um exemplo: ao utilizar uma aplicação que recomende um bom restaurante perto de você em um sábado à noite, ela precisa saber a sua localização atual e ter “pistas” sobre o seu gosto alimentar (GAO et al., 2014; MUCCINI et al., 2012; WASSERMAN, 2010).

Pesquisas recentes em testes para aplicações móveis têm focado na proposição de soluções para problemas técnicos específicos desta plataforma nas seguintes áreas (DELAMARO et al., 2016):

- **Teste Funcional e Comportamental:** valida funções de serviços, APIs web móveis, comportamentos de sistemas externos, interfaces do usuário (UIs) e seus gestos, funções

baseadas em localização, perfis de usuários, dados do sistema e dados do usuário. Muitas técnicas de teste funcional (caixa-preta) são úteis em aplicações móveis. Os testes baseados em cenários são bons exemplos. Testes baseados em interface gráfica também têm sido discutido de forma intensa em pesquisas científicas.

- **Teste Estrutural:** visa percorrer os diferentes caminhos possíveis de execução de um código para uma aplicação móvel a fim de atingir uma alta cobertura de código e garantir a qualidade da aplicação sob teste. Métodos existentes de teste estrutural (caixa branca) ainda são aplicáveis a aplicações móveis. Os itens que podem ser monitorados durante a execução do programa, aplicando essa técnica de teste, são geralmente voltados para a verificar e validar características de desempenho. Essas características podem ser referentes ao uso de recursos físicos, como bateria e memória, que impactam diretamente na velocidade da execução da aplicação móvel e seus recursos escassos em dispositivos móveis.
- **Validação de Requisitos de Qualidade de Serviço (*Quality of Service – QoS*):** avalia carga de dados, desempenho, confiabilidade/disponibilidade/privacidade, escalabilidade e vazão de dados do sistema. Pesquisas mencionam a necessidade em monitorar, medir e visualizar diversas métricas de aplicações móveis, como uso de memória, bateria ou CPU (para fins de otimização) e desempenho de redes (em vários tipos de redes de comunicação). Outro grande problema mencionado nos testes de aplicações móveis são as falhas graves (chamadas em inglês de *crashes*), que muitas vezes são intermitentes, não-determinísticas e irrecuperáveis. Assim, é necessário capturar informações suficientes sobre essas falhas para analisá-las e reproduzi-las, de modo que possam ser corrigidas.
- **Teste de Usabilidade:** avalia conteúdo e alertas na interface do usuário, fluxos e cenários de operação do usuário, mídias e apoio a interação por gestos. Testes de usabilidade, em laboratório e em campo, ajudam a melhorar a qualidade da experiência do usuário em dispositivos móveis. Esse fator tem um grande impacto no sucesso ou fracasso de uma aplicação móvel. Se usuários gostam de uma aplicação, eles baixam, começam a usá-la e avaliam positivamente nas lojas. Se não gostam, apagam, partem para outra opção imediatamente e vão às redes sociais e reclamam. Assim, uma versão de baixa qualidade e pouco atrativa aos usuários pode ter consequências devastadoras para uma aplicação móvel.
- **Teste de Compatibilidade e Conectividade:** avalia a compatibilidade de aplicações a diferentes navegadores web, plataformas e conectividade com redes de comunicação. Dispositivos móveis suportam várias opções de conectividade, então aplicações móveis

devem ser testadas com conectividades e contextos específicos. Da mesma forma, uma aplicação pode ser desenvolvida para funcionar na maior gama de dispositivos possíveis, o que resulta na necessidade de testá-la em diferentes plataformas. As dificuldades para esses testes seriam a incapacidade dos emuladores em simular o comportamento real dos dispositivos, ferramentas com apoio limitado a recursos importantes para esses testes, tais como mobilidade, serviços de localização, sensores e diferentes tipos de entrada, e ferramentas que permitam migrar testes entre plataformas.

Analisando o cenário atual de aplicações móveis percebe-se a infinidade de ferramentas, métodos e técnicas de teste aplicadas, por isso há certa dificuldade na definição de quais serão aplicados a certos tipos de aplicações. Dessa forma foi identificada a necessidade de agrupar métodos, técnicas, ferramentas e processos e agrupá-los de acordo com o tipo de aplicação e característica a ser testada.

## **2.7. Trabalhos Relacionados**

Na literatura técnica não foram identificados estudos que abordem linhas de processo de software para o teste de aplicações móveis. Assim, serão considerados trabalhos que abordem linhas de processo de software para outras áreas. A seguir são descritos alguns destes trabalhos.

Armbrust *et al.* (ARMBRUST et al., 2008) adapta processos de Linha de Produto de Software para serem aplicados a processos de software. Para definição do escopo da LPrS, analisou-se os produtos e projetos da empresa, os processos, priorizou-se os atributos e determinação de um escopo. No trabalho, foi realizado um estudo de caso em alguns projetos no *Japan Aerospace Exploration Agency* (JAXA) envolvendo processos de desenvolvimento de aplicações e processos gerenciais, onde foi realizada a aplicação para apenas alguns atributos, sem comparação dois-a-dois. Resultados preliminares mostraram a viabilidade da proposta, mas não sua aplicabilidade geral.

Garcia *et al.* (GARCIA et al., 2015) propõem uma LPrS para desenvolvimento de aplicações usando SOA e BPM. Foi abordada a orientação a serviços e propostos processos de desenvolvimentos voltado ao reuso em orientação a serviços. ALPrS proposta foi chamada de SOPLA e explorava os modelos existentes para melhorar o processo de desenvolvimento desta natureza e integrar paradigmas, ferramentas e práticas envolvidas. Foi adaptada a abordagem proposta por (ALEIXO et al., 2011), desenvolvida uma ferramenta para transformação do modelo do EPF para notação BPMN e integração de atividades de modelos de gerenciamento de processos de negócio com o processo de desenvolvimento de software. A avaliação da LPrS foi realizada por especialistas, considerando que a aplicação da LPrS em uma organização exigia muito tempo de observação e treinamento envolvido tornando inviável para o tempo estimado da pesquisa de

apenas dois anos.

Magdaleno *et al.* (MAGDALENO et al., 2015) propõem a criação de uma Linha de Processo de Software no contexto de uma companhia de gás e óleo brasileira. Para auxiliar na composição da LPrS foram utilizados o *Context-Based Process Line Engineering* (CBPL) (MAGDALENO et al., 2012) e COMPOOTIM (MAGDALENO et al., 2014). No trabalho a criação da LPrS é tratada como um estudo de caso, utilizando os modelos de processos organizacionais e padrão como base, a partir dos dados foi elaborado um documento contendo as atividades, papéis e artefatos do processo de software de forma detalhada. Segundo o documento existiam 12 metodologias de desenvolvimento na organização, sendo escolhida a Orientada a Objeto para composição da linha, os subprocessos de gerenciamento de processo tradicional, gerenciamento de processos com metodologia ágil, teste e gerenciamento de configuração também foram adotados. Após a composição da linha a mesma foi submetida a avaliação por especialistas, por meio de um *survey*, que continha uma lista dos elementos dos processos e modelo de características (*feature model*). Ao todo foram realizadas quatro sessões individuais, onde cada uma tinha uma duração de 01:20 h. Os defeitos identificados foram classificados por nível de severidade sendo: 25 defeitos de baixa severidade, 15 como média e 1 como alto.

A partir dos conceitos apresentados, foi concebida uma LPrS para apoiar o teste de aplicações móveis, sendo detalhada no próximo capítulo.

## 3. MTPL: LINHA DE PROCESSO DE SOFTWARE PARA TESTE DE APLICAÇÕES MÓVEIS

### 3.1. Introdução

A proposta MTPL – *Mobile Testing Process Line*, é uma linha de processos de software para reduzir o esforço do gerente/líder de testes, na qual, com a utilização da seleção de características, este será capaz de gerar um novo processo para o teste de aplicações móveis. Os passos que o gerente/líder fará por meio da linha são:

1. Ao iniciar um novo projeto, deve-se analisar suas características;
2. Selecionar as características desejadas utilizando a ferramenta;
3. Gerar uma instância do processo derivando da linha somente as características selecionadas. Este passo é realizado de forma automática pela ferramenta.
4. A instância do processo é apresentada por meio de relatório ao usuário.

O passo inicial realizado para a concepção da linha foi o levantamento das características e requisitos para a definição do escopo da linha. Ao analisar o cenário de teste de aplicações móveis percebe-se a necessidade de um agrupamento em processos a infinidade de métodos, técnicas, ferramentas e tipos de teste publicados na literatura. Entretanto, esses processos possuem muitas características próprias de cada aplicação respeitando suas limitações e necessidades.

A concepção da linha enfatizou um escopo direcionado as fases técnicas de um processo de teste, ou seja, foram detalhadas as fases de Design & Implementação e Execução. A documentação da linha deteve-se ao modelo de características (*feature model*) disponível no Apêndice C, os elementos de maior granularidade do SPEM (fases e atividades) e detalhando os de menor granularidade (tarefas). As atividades relacionadas ao planejamento e análise & encerramento não fazem parte do escopo deste trabalho.

### 3.2. Conteúdo da MTPL

O conteúdo da ferramenta é a parte conceitual utilizada para definição dos processos de teste para aplicações móveis. Dentro da linha proposta foram utilizados alguns papéis tradicionais do teste de software. A lista de papéis a seguir está vinculada às atividades como responsáveis e participantes.

- Gerente de testes: O papel envolve defesa da qualidade e dos testes, planejamento e

gerenciamento de recursos e resolução de problemas que representam um obstáculo para o esforço de teste. Isso inclui:

- o Negociar a finalidade e os produtos provenientes do esforço de teste;
  - o Assegurar o planejamento e o gerenciamento apropriado dos recursos de teste;
  - o Avaliar o andamento e a eficácia do esforço de teste;
  - o Assegurar o nível apropriado de qualidade mediante a correção de defeitos importantes;
  - o Assegurar um nível apropriado de enfoque na testabilidade durante o processo de desenvolvimento de software.
- **Projetista de testes ou Designer de testes:** Responsável por definir a abordagem de teste e assegurar sua correta implementação. O papel envolve identificar as técnicas, ferramentas e diretrizes apropriadas para implementar os testes necessários e dar orientação sobre os correspondentes requisitos de recursos para o esforço de teste. Este papel é responsável por:
    - o Identificar e descrever as técnicas de teste apropriadas;
    - o Identificar as ferramentas de suporte apropriado;
    - o Definir e manter a arquitetura de automatização de teste;
    - o Especificar e verificar as configurações do ambiente de teste obrigatórias;
    - o Verificar e avaliar a abordagem de teste.
  - **Testador:** O papel testador é responsável pelas atividades centrais do esforço de teste, que envolve conduzir os testes necessários e registrar os resultados desses itens. Isso inclui:
    - o Identificar a abordagem de implementação mais apropriada para um dado teste;
    - o Implementar testes individuais;
    - o Configurar e executar testes;
    - o Registrar os resultados e verificar a execução dos testes;
    - o Analisar erros de execução e recuperar-se.
  - **Desenvolvedor:** responsável pela implementação e correção de erros/defeitos no sistema. Seu foco é desenvolver novas soluções e pôr em prática todas as necessidades impostas pelo analista de sistemas.

A definição da entrada e saída de cada atividade é realizada por meio dos produtos de

trabalho. Para a linha apresentada, foram propostos os seguintes artefatos extraídos da norma ISO/IEEE 29119-2 (ISO/IEEE 29119-2, 2013):

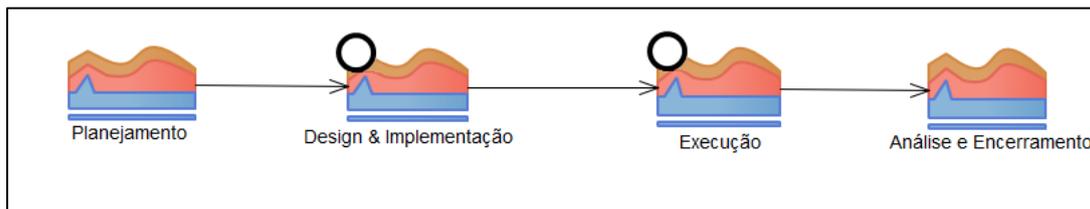
- Plano de testes: A definição das metas e dos objetivos dos testes no escopo da iteração (ou projeto), os itens-alvo, a abordagem adotada, os recursos necessários e os produtos que serão liberados.
- Projeto de testes: É um registro documentado de qualquer um destes itens: decisões de controle e execução do processo, padrões que devem servir de base ou orientações de boas práticas que geralmente devem ser seguidas pelos participantes de um projeto específico.
- Caso de teste: É a definição (geralmente formal) de um conjunto específico de entradas de teste, condições de execução e resultados esperados, identificados com a finalidade de avaliar um determinado aspecto de um item de teste-alvo.
- Procedimento de teste: Um procedimento de teste é uma série de instruções detalhadas para a configuração, execução, e avaliações de resultados para uma série de casos de teste.
- Relatório de incidentes de teste: Toda e qualquer discrepância entre o resultado esperado e o encontrado na execução dos casos de teste devem ser reportados para o time de desenvolvimento com o máximo de detalhes possíveis. Nesse caso, é utilizado o relatório de incidentes de teste no qual são registrados todos os defeitos encontrados durante a fase de execução de teste.
- Histórico de testes: É um conjunto de dados brutos de saída capturados durante uma execução exclusiva de um ou mais testes, geralmente representando a saída resultante da execução de um conjunto de testes para uma única execução do ciclo de teste.
- Lições aprendidas: Documento contendo a experiência adquirida ao longo de um projeto. Lições aprendidas registram os problemas ocorridos, e como eles foram, ou deveriam ter sido, resolvidos.
- Resumo dos testes: Um conjunto de informações resumidas determinadas pela análise de um ou mais registros de teste e solicitações de mudança, que permitem uma avaliação relativamente detalhada da qualidade dos itens de teste-alvo e do status de cada esforço de teste. Muitas vezes, é também consultado como um repositório maior de vários resultados do teste.

Foram extraídas uma série de ferramentas componentes dos processos especificados pela LPrS. A lista de ferramentas encontra-se disponível no Apêndice D.

### 3.3. Processo da Linha de Processo de Software

Inicialmente foram definidas as fases do ciclo de vida para posteriormente identificar as atividades e variabilidades. Essas fases foram baseadas no mapeamento sistemático apresentado na seção 2.6 e na norma ISO/IEEE 29119-2 (ISO/IEEE 29119-2, 2013).

O ciclo de vida foi representado por meio de fases consideradas obrigatórias (Figura 25). As fases do ciclo de vida propostas são:



**Figura 25.** Ciclo de vida de teste de software

- Planejamento de teste: Definição de uma proposta de testes baseada nas expectativas do cliente em relação à prazos, custos e qualidade esperada, possibilitando dimensionar a equipe e estabelecer um esforço de acordo com as necessidades apontadas pelo cliente.
- Design & implementação: Identificação dos casos de testes que deverão ser construídos e modificados em função das mudanças solicitadas pelo cliente, bem como pelo próprio aperfeiçoamento do processo de testes (ampliação da cobertura). Identificação de todos os elementos necessários para a implementação de cada caso de testes especificado. Fazem parte desta etapa a modelagem dos testes e a definição dos critérios de tratamento de arquivos;
- Execução: Caracterizada pela execução e conferência dos testes planejados, de forma a garantir que o comportamento do aplicativo permanece em “conformidade” com os requisitos contratados pelo cliente.
- Análise & encerramento: Caracterizada pela análise e confirmação dos resultados relatados durante a fase de execução dos testes. Os resultados em “não-conformidade” deverão ser “confirmados” e “detalhados” para que a fábrica de software realize as correções necessárias. Já os casos de teste progressivos em “conformidade” deverão ter seu resultado “POSITIVO” reconfirmado e seu “*baseline*” atualizado.

#### 3.3.1. Fase de planejamento

Ao longo dessa fase é realizado o planejamento do processo de teste a ser seguido para um projeto específico, onde são estimados custos, cronograma e recursos, e são definidos os itens a serem testados, as estratégias, métodos e técnicas de teste a serem adotadas, caracterizando-se

um critério para aceitação do software testado.

Esta fase deve ser realizada pelo Gerente de Testes e envolve muitas tomadas de decisões que devem ser analisadas buscando-se encontrar as melhores opções, dado o contexto de um projeto. Pontos chaves do planejamento dos testes estão relacionados à forma como os testes serão realizados, qual a equipe de teste, os recursos disponíveis, restrições de tempo e os riscos associados (ABRAN et al., 2004).

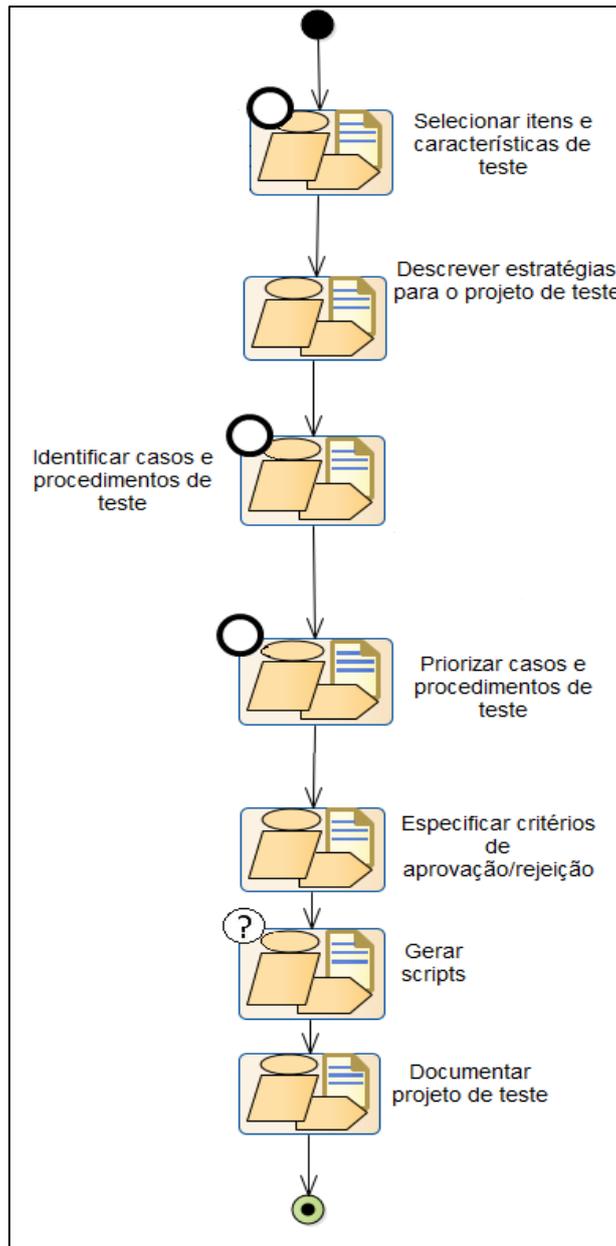
Por não se tratar do contexto da pesquisa não foram analisados pontos de variação e variabilidades para essa fase, sendo apenas listadas suas atividades.

- Caracterizar testes;
- Definir atividade / tarefas de teste;
- Definir tipos e técnicas de teste;
- Identificar e analisar riscos;
- Projetar estratégias de teste;
- Documentar plano de teste;
- Validar plano de teste.

### **3.3.2. Design & Implementação**

De acordo com o IEEE Std 829 (ISO/IEEE 829-2008, 2008), a fase de design & implementação visa à especificação mais detalhada das abordagens a serem seguidas durante a realização dos testes e que foram identificadas durante a fase de planejamento dos testes. Além disso, essa fase é responsável pela identificação do conjunto de casos e procedimentos de teste a serem executados para avaliação do software. Devem ser especificados ainda todos os casos de teste identificados, isto é, devem ser descritos seus valores de entrada, resultados esperados, recursos necessários para a sua execução, suas restrições e dependências com outros casos de teste. As atividades que devem ser realizadas durante a fase de design & implementação de teste de software devem ser conduzidas pelo Projetista de Testes.

Para a referida fase, por estar incluída nas fases técnicas do processo de teste de software, foram analisados pontos de variação e variabilidades (Figura 26), suas atividades detalhadas podem ser observadas nas Tabelas 9 a 15 (DIAS-NETO e TRAVASSOS, 2006).



**Figura 26.** Fase de design & implementação

**Tabela 9.** Atividade selecionar itens e características de teste

<b>Nome</b>	Selecionar itens e características de teste
<b>Descrição</b>	Durante esta atividade, devem ser selecionados os itens de teste e as características (ou combinações de características) que serão avaliadas no projeto dos testes em questão. Os testes para cada característica ou combinação de características devem ser projetados isoladamente.
<b>Tipo</b>	Atividade
<b>Variabilidade</b>	Ponto de variação obrigatório
<b>Variantes</b>	<ul style="list-style-type: none"><li>● Itens e características funcionais / comportamentais;</li><li>● Itens e características de usabilidade;</li><li>● Itens e características de qualidade de serviço;</li><li>● Itens e características de compatibilidade e conectividade.</li></ul>
<b>Responsável</b>	Projetista de testes

**Tabela 10.** Atividade descrever estratégias para o projeto de testes

<b>Nome</b>	Descrever estratégias para o projeto de testes
<b>Descrição</b>	Durante esta atividade, deve ser especificada com detalhes a estratégia descrita no plano de teste para avaliar o item de teste e a característica selecionados na atividade anterior, incluindo as técnicas e ferramentas específicas a serem utilizadas. Isso consiste em um detalhamento do que foi descrito superficialmente durante o planejamento dos testes. Além disso, deverá ser identificada a forma como os resultados dos testes serão analisados.
<b>Tipo</b>	Atividade
<b>Responsável</b>	Projetista de testes

**Tabela 11.** Atividade identificar casos de teste e procedimentos

<b>Nome</b>	Identificar casos de teste e procedimentos
<b>Descrição</b>	Casos de Teste são pontos essenciais na execução dos testes em software. Testar exaustivamente ou testar todas as combinações de entradas possíveis é inviável (MYERS, 1979). Com isso, a seleção de casos de teste corretos é um fator essencial para o sucesso da atividade de teste. Durante esta atividade, deve ser identificado o conjunto de casos e procedimentos de teste para avaliar o item de teste e a característica selecionada para o projeto de teste em questão.
<b>Tipo</b>	Atividade
<b>Variabilidade</b>	Ponto de variação obrigatório
<b>Variantes</b>	<p>Itens e características funcionais / comportamentais</p> <ul style="list-style-type: none"><li>• Abordagem de teste de gestos;</li><li>• Abordagem de teste de interações;</li><li>• Abordagem de teste de transações;</li><li>• Abordagem de teste de eventos;</li></ul> <p>Itens e características de usabilidade</p> <ul style="list-style-type: none"><li>• Abordagem empírica;<ul style="list-style-type: none"><li>○ Observação de campo;</li><li>○ Entrevista e questionário;</li><li>○ Experimento controlado;</li><li>○ Grupo focal;</li><li>○ Registro verbal;</li><li>○ Registro de uso.</li></ul></li><li>• Abordagem não-empírica<ul style="list-style-type: none"><li>○ Avaliação de perito;</li><li>○ Lista de verificação;</li><li>○ Percurso cognitivo;</li></ul></li></ul>

**Tabela 12** (Continuação)

	<ul style="list-style-type: none"><li>○ Avaliação heurística;</li></ul> Itens e características de qualidade de serviço <ul style="list-style-type: none"><li>• Desempenho / confiabilidade;</li><li>• Segurança e privacidade.</li></ul> Itens e características de compatibilidade e conectividade <ul style="list-style-type: none"><li>• Mobilidade e conectividade</li><li>• Interoperabilidade.</li></ul>
<b>Responsável</b>	Projetista de testes.

**Tabela 13.** Atividade priorizar casos de teste e procedimentos

<b>Nome</b>	Priorizar casos de teste e procedimentos
<b>Descrição</b>	Segundo (DUSTIN, 2002), o planejamento de casos e procedimentos de teste deve ser baseado na ordem de prioridade. A ordem de prioridade dos casos e procedimentos de teste deve ser definida baseando-se em alguns fatores, como: os riscos associados aos itens de teste definidos durante a fase de planejamento de testes e o cronograma dos testes (DUSTIN, 2002). Durante esta atividade deve ser especificada a ordem de prioridade dos casos e procedimentos de teste levando-se em consideração esses ou outros aspectos definidos.
<b>Tipo</b>	Atividade
<b>Variabilidade</b>	Ponto de variação obrigatório
<b>Variantes</b>	Não cobertas pelo escopo do trabalho, por se tratar de um tema de pesquisa amplo e complexo envolvendo contextos multidisciplinares.
<b>Responsável</b>	Projetista de testes

**Tabela 14.** Atividade especificar critérios de aprovação / rejeição

<b>Nome</b>	Especificar critérios de aprovação / rejeição
<b>Descrição</b>	Durante esta atividade, devem ser especificados os critérios a serem utilizados para determinar, exclusivamente, quando a característica (ou a combinação de características) selecionada para o projeto de teste será aprovada ou rejeitada.
<b>Tipo</b>	Atividade
<b>Responsável</b>	Projetista de testes

**Tabela 15.** Atividade gerar scripts

<b>Nome</b>	Gerar scripts
<b>Descrição</b>	Durante esta atividade, devem ser gerados ser gerados os scripts de teste a partir dos casos e procedimentos de teste especificados. Essa atividade só será realizada caso sejam executados testes automatizados.
<b>Tipo</b>	Atividade
<b>Variabilidade</b>	Opcional
<b>Responsável</b>	Projetista de testes

**Tabela 16.** Atividade documentar projeto de testes

<b>Nome</b>	Documentar projeto de testes
<b>Descrição</b>	Refina a abordagem de teste a ser seguida e que foi apresentada no plano de teste para avaliação de um (ou vários) item de teste em relação a uma característica (ou uma combinação de características) de teste que tenha sido especificada. Este documento também identifica os casos e os procedimentos de teste e apresenta os critérios para aprovação do (s) item (ou itens) de teste avaliado(s) neste projeto de teste específico. Cada característica (ou combinação de características) a ser avaliada durante os testes deve possuir um documento de especificação de projeto de teste específico
<b>Tipo</b>	Atividade
<b>Responsável</b>	Projetista de testes

### 3.3.3. Execução

O objetivo dessa fase é a execução dos testes a partir do que foi estabelecido durante o planejamento, monitorando as atividades realizadas, registrando os incidentes detectados e apresentando ao final os resultados dos testes. Ao final serão estabelecidos os seguintes documentos: Histórico dos testes, relatório(s) de incidente de teste e relatório de resumo de teste, além do pacote com os dados dos testes realizados.

Para a referida fase, por estar incluída nas fases técnicas do processo de teste de software, foram analisados pontos de variação e variabilidades (Figura 27), suas atividades detalhadas podem ser observadas nas Tabelas 16 a 20 (DIAS-NETO e TRAVASSOS, 2006).

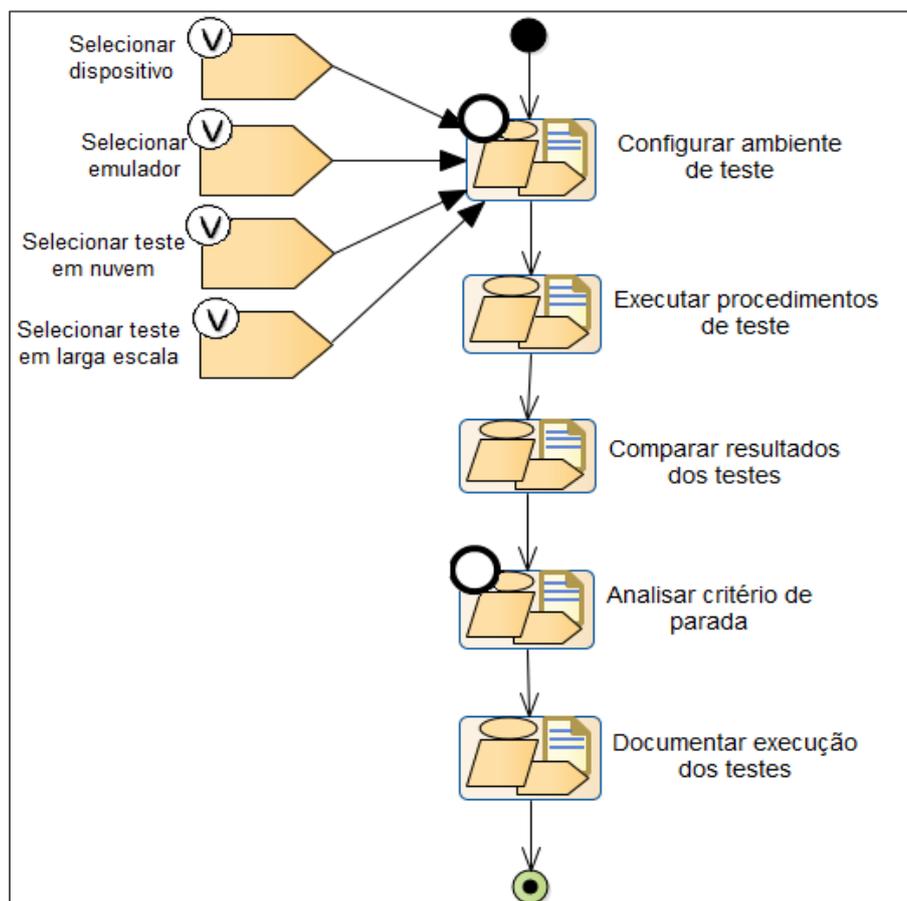


Figura 27. Fase de execução

**Tabela 17.** Atividade configurar ambiente de testes

<b>Nome</b>	Configurar ambiente de testes
<b>Descrição</b>	O ambiente de teste deve ser compatível com ambiente de desenvolvimento de software. Ele deve facilitar a execução e controle dos casos e procedimentos de teste, bem como registrar os resultados obtidos. Durante esta atividade, o ambiente de teste deve ser ajustado para a realização dos testes previamente planejados.
<b>Tipo</b>	Atividade
<b>Variabilidade</b>	Ponto de variação obrigatório
<b>Variantes</b>	<ul style="list-style-type: none"><li>• Selecionar dispositivo;</li><li>• Selecionar emulador;</li><li>• Selecionar teste em nuvem;</li><li>• Selecionar teste em larga escala.</li></ul>
<b>Responsável</b>	Testador

**Tabela 18.** Atividade executar procedimentos de teste

<b>Nome</b>	Executar procedimentos de teste
<b>Descrição</b>	Durante esta atividade devem ser executados os procedimentos de teste previamente definidos e para cada procedimento de teste a ser realizado será realizada uma comparação entre os resultados obtidos por meio da execução e o resultado esperado, definido durante o processo de planejamento. Cada evento realizado deve ser registrado no documento histórico dos testes.
<b>Tipo</b>	Atividade
<b>Responsável</b>	Testador

**Tabela 19.** Comparar resultados dos testes

<b>Nome</b>	Comparar resultados dos testes
<b>Descrição</b>	A atividade consiste na comparação dos resultados esperados com cada caso de teste do procedimento de testes. Os resultados esperados dos casos de teste executados nos procedimentos devem ser determinados previamente.
<b>Tipo</b>	Atividade
<b>Responsável</b>	Testador

**Tabela 20.** Atividade analisar critério de parada

<b>Nome</b>	Analisar critério de parada
<b>Descrição</b>	Essa atividade tem o objetivo de definir critérios que podem ser usados para definir em que momento as atividades de teste podem parar em um projeto de desenvolvimento de software sem prejudicar o atendimento aos requisitos de qualidade. Frequentemente critérios de parada são baseados em modelos de confiabilidade nos quais se propõe prever ou determinar a confiabilidade do software.
<b>Tipo</b>	Atividade
<b>Variabilidade</b>	Ponto de variação obrigatório
<b>Variantes</b>	Não cobertos pelo escopo do trabalho
<b>Responsável</b>	Testador

**Tabela 21.** Atividade documentar execução dos testes

<b>Nome</b>	Documentar execução dos testes
<b>Descrição</b>	A execução dos testes deve ser documentada conforme especificado no plano de teste.
<b>Tipo</b>	Atividade
<b>Responsável</b>	Testador

### **3.3.4. Análise e encerramento**

Os resultados dos testes devem ser avaliados para determinar se os testes obtiveram sucesso, além de obter medidas de teste específicas (ABRAN et al., 2004). Na maioria dos casos, “sucesso” significa que o sistema funcionou conforme o esperado, e não apresentou resultados inesperados. A análise dos resultados dos testes permite a uma organização prover um entendimento sobre o seu processo de desenvolvimento identificando pontos de falhas durante o desenvolvimento, e consequentemente prover melhorias no seu processo.

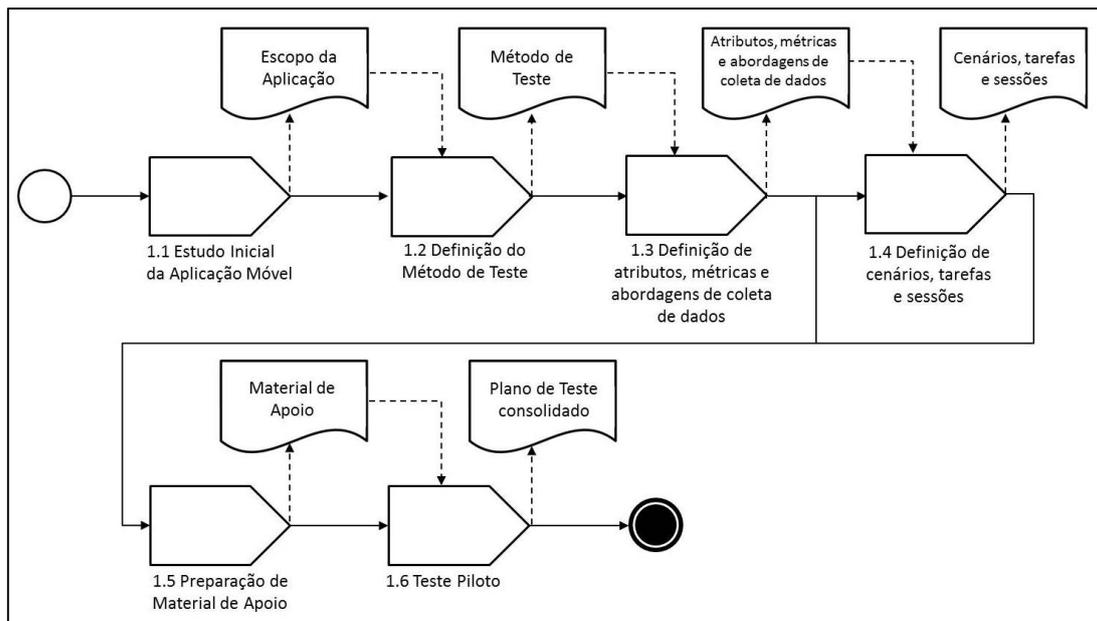
As atividades a serem realizadas durante a definição dos procedimentos de teste devem ser conduzidas pelo Gerente de Teste.

Por não se tratar do contexto da pesquisa não foram analisados pontos de variação e variabilidades para essa fase, sendo assim é apresentada apenas uma lista das atividades que compõem a fase.

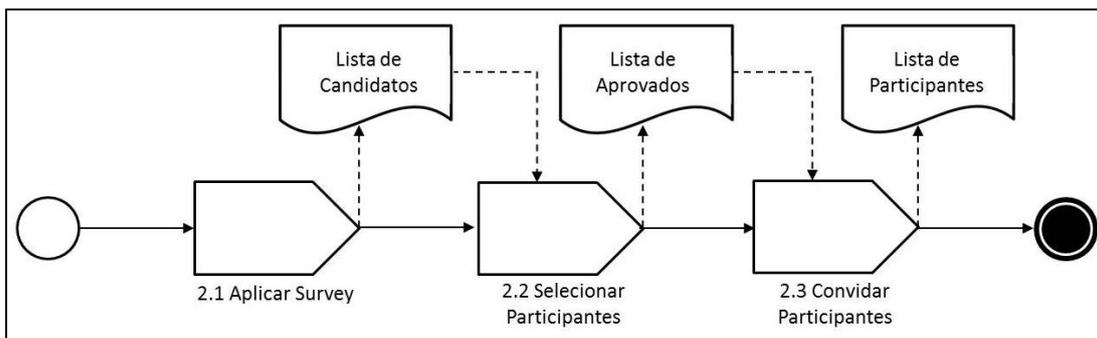
- Gerar relatório de testes;
- Restaurar ambiente de testes;
- Identificar lições aprendidas;
- Documentar análise e encerramento.

### **3.3.5. Processos para teste de usabilidade**

Por possuírem características específicas foi necessário criar um novo processo de design & implementação para os itens e características de usabilidade. O processo foi derivado de (MENDES e DIAS-NETO, 2016) e pode ser observado nas Figuras 28 e 29.



**Figura 28.** Processo de preparação de testes de usabilidade. (MENDES e DIAS-NETO, 2016)



**Figura 29.** Processo de recrutamento de participantes para testes de usabilidade. (MENDES e DIAS-NETO, 2016)

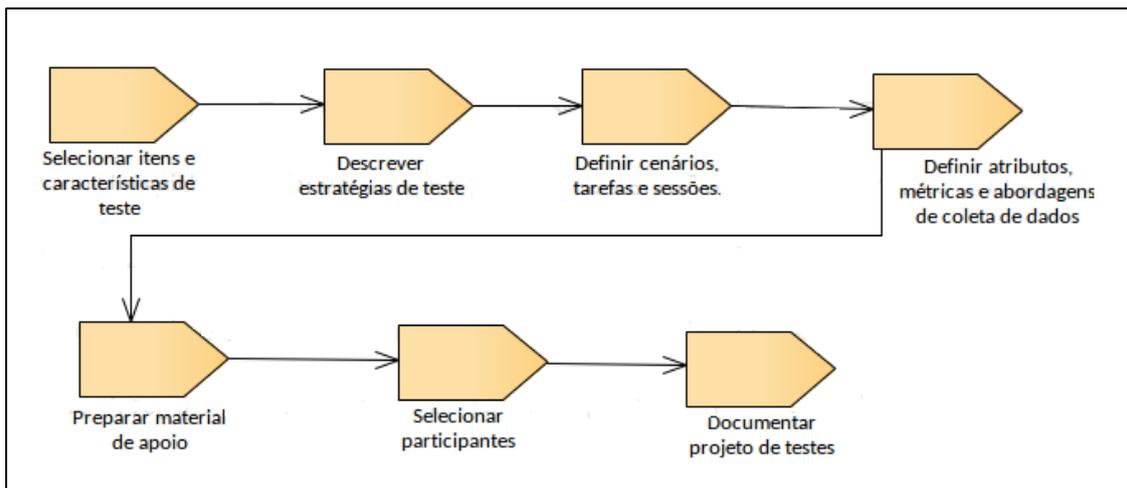
No processo derivado foram mescladas as fases de preparação e recrutamento de participantes na fase de Design & Implementação. Analisando a fase de preparação do processo original, foram mantidas as atividades de 1.4 - Definição de cenários, tarefas e sessões, 1.3 - Definição de atributos, métricas e abordagens de coleta de dados e 1.5 - Preparação de material de apoio. As demais atividades foram desconsideradas por já estarem inclusas na fase de Planejamento do processo proposto. A atividade 2.2 – Selecionar participantes da fase de recrutamento também foi incluída no processo adaptado, tendo em vista que o mesmo não englobava preocupações quanto a abordagens de teste de usabilidade empíricos.

### 3.3.5.1. Design & Implementação

De acordo com o IEEE Std 829 (ISO/IEEE 829-2008, 2008), a fase de design & implementação visa à especificação mais detalhada das abordagens a serem seguidas durante a realização dos testes e que foram identificadas durante a fase de planejamento dos testes. Além disso, essa fase é responsável pela identificação do conjunto de casos e procedimentos de teste a serem executados para avaliação do software.

As atividades que devem ser realizadas durante a fase de design & implementação de testes de usabilidade devem ser conduzidas pelo Projetista de Testes.

Para a referida fase, por estar incluída nas fases técnicas do processo de teste de software, foram analisados pontos de variação e variabilidades (veja Figura 30), suas atividades detalhadas podem ser observadas nas Tabelas 21 a 27.



**Figura 30.** Design & Implementação para teste de Usabilidade

**Tabela 22.** Atividade selecionar itens e características de teste - Usabilidade

<b>Nome</b>	Selecionar itens e características de teste
<b>Descrição</b>	Durante esta atividade, devem ser selecionados os itens de teste e as características (ou combinações de características) que serão avaliadas no projeto dos testes em questão. Os testes para cada característica ou combinação de características devem ser projetados isoladamente.
<b>Tipo</b>	Atividade
<b>Responsável</b>	Projetista de teste

**Tabela 23.** Atividade descrever estratégias de teste - Usabilidade

<b>Nome</b>	Descrever estratégias de teste
<b>Descrição</b>	<p>Devem ser estabelecidas as estratégias de teste que serão adotadas especificadas no Plano de Teste, saída da fase de planejamento de teste.</p> <p>Uma estratégia de testes descreve a abordagem geral e os objetivos das atividades de teste. Ela deve contemplar os níveis ou fases de teste, os tipos de testes a serem realizados e as técnicas para sua execução. A estratégia de testes também deve descrever com clareza os critérios para a conclusão dos testes e os critérios de sucesso a serem usados.</p>
<b>Tipo</b>	Atividade
<b>Responsável</b>	Projetista de teste

**Tabela 24.** Atividade definir cenários, tarefas e sessões

<b>Nome</b>	Definir cenários, tarefas e sessões
<b>Descrição</b>	<p>Devem ser identificadas quais tarefas deverão mantidas, atualizadas ou excluídas, e se há necessidade de criar novas tarefas. Baseado nesses resultados é possível reformular a lista de tarefas e cenários instanciando para outra plataforma.</p>
<b>Tipo</b>	Atividade
<b>Responsável</b>	Projetista de teste

**Tabela 25.** Atividade definir atributos, métricas e abordagens de coleta de dados

<b>Nome</b>	Definir atributos, métricas e abordagens de coleta de dados
<b>Descrição</b>	<p>A partir da lista de tarefas e cenários provenientes da atividade definir cenários, tarefas e sessões é possível especificar métricas que podem ser utilizadas para comparar resultados entre os dispositivos/plataformas selecionadas. Alguns exemplos de métricas são:</p>

**Tabela 26** (Continuação)

	<ul style="list-style-type: none"> <li>• Caminhos possíveis para completar um cenário/tarefa;</li> <li>• Número de teclas ou toques na tela para completar o cenário/tarefa.</li> </ul>
<b>Tipo</b>	Atividade
<b>Responsável</b>	Projetista de teste

**Tabela 27.** Atividade preparar material de apoio

<b>Nome</b>	Preparar material de apoio
<b>Descrição</b>	A partir dos cenários, métricas e tarefas definidas devem ser elaborados documentos auxiliares, como questionários de caracterização de candidato, preparação de dispositivos, e qualquer outro material especificado na etapa de planejamento.
<b>Tipo</b>	Atividade
<b>Responsável</b>	Projetista de teste

**Tabela 28.** Atividade selecionar participantes

<b>Nome</b>	Selecionar participantes
<b>Descrição</b>	<p>Essa atividade tem o objetivo de selecionar participantes para o teste em todas as plataformas. Inicialmente pode-se aplicar um <i>survey</i> entre os candidatos para extrair características específicas dos participantes. Depois são identificados quais participantes atendem os requisitos para os testes. Ao final esses participantes devem ser contatados para o agendamento dos testes. Existem boas práticas para essa atividade, são elas:</p> <ul style="list-style-type: none"> <li>• Escolher a mesma quantidade de participantes para cada plataforma a ser testada com um perfil e conhecimento a respeito da plataforma similar;</li> <li>• Balancear a distribuição de participantes em cada plataforma de acordo com sua expertise.</li> </ul>

**Tabela 29.** (Continuação)

<b>Tipo</b>	Atividade
<b>Responsável</b>	Projetista de teste

**Tabela 30.** Atividade documentar projeto de testes

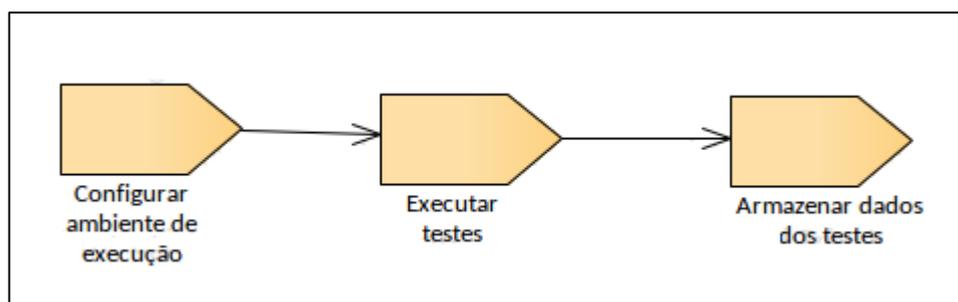
<b>Nome</b>	Documentar projeto de testes
<b>Descrição</b>	Refina a abordagem de teste a ser seguida e que foi apresentada no plano de teste para avaliação de um (ou vários) item de teste em relação a uma característica (ou uma combinação de características) de teste que tenha sido especificada. Este documento também identifica os casos e os procedimentos de teste e apresenta os critérios para aprovação do(s) item (ou itens) de teste avaliado(s) neste projeto de teste específico. Cada característica (ou combinação de características) a ser avaliada durante os testes deve possuir um documento de especificação de projeto de teste específico
<b>Tipo</b>	Atividade
<b>Responsável</b>	Projetista de teste

### 3.3.5.2. Execução

O objetivo dessa fase é a execução dos testes a partir do que foi estabelecido durante o planejamento, monitorando as atividades realizadas, registrando os incidentes detectados e apresentando ao final os resultados dos testes.

Ao final serão estabelecidos os seguintes documentos: Histórico dos testes, relatório(s) de incidente de teste e relatório de resumo de teste, além do pacote com os dados dos testes realizados.

Para a referida fase, por estar incluída nas fases técnicas do processo de teste de software, foram analisados pontos de variação e variabilidades (Figura 31), suas atividades detalhadas podem ser observadas nas tabelas 28 a 30 (MENDES e DIAS-NETO, 2016).



**Figura 31.** Execução para teste de Usabilidade.

**Tabela 31.** Atividade configurar ambiente de execução - usabilidade

<b>Nome</b>	Configurar ambiente de execução
<b>Descrição</b>	O ambiente de teste deve ser compatível com ambiente de desenvolvimento de software. Ele deve facilitar a execução e controle das sessões e cenários, bem como registrar os resultados obtidos. Durante esta atividade, o ambiente de teste deve ser ajustado para a realização dos testes previamente planejados.
<b>Tipo</b>	Atividade
<b>Responsável</b>	Testador

**Tabela 32.** Atividade executar testes - usabilidade

<b>Nome</b>	Executar testes
<b>Descrição</b>	Durante esta atividade devem ser executados os testes previamente definidos. Cada evento realizado deve ser registrado e armazenado.
<b>Tipo</b>	Atividade
<b>Responsável</b>	Testador

**Tabela 33.** Atividade armazenar dados dos testes

<b>Nome</b>	Armazenar dados dos testes
<b>Descrição</b>	Os testes devem ser executados e os dados dos vídeos/áudios/survey respondidos devem ser armazenados individualmente para cada plataforma em forma de planilhas ou outros documentos.

**Tabela 34.** (Continuação)

<b>Tipo</b>	Atividade
<b>Responsável</b>	Testador

### **3.4. Apoio computacional à linha proposta**

As ferramentas para instanciação dos processos específicos em LPrS possuem limitações na instanciação de processos e interface, sendo necessário ao utilizador conhecimento prévio a respeito de LPrS e manipulação do sistema. Sendo assim, para realizar a instanciação do processo de teste foi implementado uma ferramenta baseado em questionário, utilizando como base a plataforma *LimeSurvey*. A ferramenta foi batizado de MTPLT (*Mobile Testing Process Line Tool*) e consiste em uma série de perguntas feitas ao utilizador, buscando identificar os requisitos necessários para instanciação do processo. A ferramenta é uma aplicação web, sendo necessário apenas um navegador e acesso à internet para sua utilização.

#### **3.4.1. Funcionamento da MTPLT**

O principal diferencial da ferramenta é o fato de instanciar processos de teste para aplicações móveis de forma automatizada após a inserção dos requisitos de teste. Como pré-requisito, ao iniciar a aplicação o usuário precisa ter conhecimento a respeito dos parâmetros utilizados nos testes e requisitos básicos de funcionamento da aplicação. Ambas as informações são fornecidas no plano de teste. Por esse motivo, a ferramenta é direcionada as fases técnicas do processo de teste (Design & Implementação e Execução). Esses processos instanciados são baseados no agrupamento das possíveis variações entre os componentes da LPrS, alinhados aos requisitos especificados. É importante ressaltar que podem existir outras instanciações dos processos, se levados em consideração requisitos não mapeados na LPrS proposta. Ao longo desta seção serão descritos os passos para instanciação do processo utilizando a ferramenta.

##### **3.4.1.1. Selecionar estratégia de teste**

Para iniciar o uso da ferramenta, o usuário deve selecionar a estratégia de teste a ser adotada, como mostrado na Figura 32. A estratégia de teste foi definida na fase de Planejamento de teste e está registrada no plano de testes.

\* Quais técnicas e estratégias de teste devem ser utilizadas ? (Caso tenha dúvidas a respeito das definições clique [aqui](#))

ⓘ Por favor, escolha no mínimo uma resposta

Testes funcionais / comportamentais  
(Garantem que as funcionalidades estejam de acordo com os requisitos)

Teste de usabilidade  
(Ajuda a melhorar a experiência do usuário)

Teste de qualidade de serviço  
(Avalia carga de dados, desempenho, confiabilidade, etc)

Teste de compatibilidade e conectividade  
(Avalia a compatibilidade da aplicação em diferentes dispositivos e conexões de rede)

**Figura 32.** Tela selecionar estratégias de teste

Para definição das estratégias possíveis, foi realizada uma busca informal na literatura técnica utilizando como base o material publicado em (DELAMARO et al, 2016), com o objetivo de selecionar as principais estratégias aplicadas na academia e indústria para teste de aplicações móveis. Algumas estratégias tradicionais foram retiradas por não representar uma aplicação relevante a aplicações móveis. Durante esse passo podem ser selecionadas uma ou mais estratégias de teste, contendo ainda um guia de ajuda caso surjam dúvidas a respeito das estratégias apresentadas.

### **3.4.1.2. Selecionar abordagens de teste**

Como segundo passo no uso da ferramenta devem ser selecionadas as abordagens de teste a serem utilizadas (Figura 33). As abordagens de teste são selecionadas de acordo com a estratégia escolhida no passo anterior. A abordagem de teste foi definida na fase de Planejamento de teste e está registrada no plano de testes.

\* Quais abordagens funcionais / comportamentais serão utilizadas? (Para dúvidas a respeito das definições clique [aqui](#))

Teste de gestos (*Mobile gestures*)  
(Testar gestos de usuário aceitos pelo dispositivo móvel são executados corretamente na aplicação).

Teste de interações do usuário  
(Teste através da interface da aplicação considerando o perfil do usuário).

Teste de transações  
(Visa validar os requisitos e as transações de negócio na aplicação).

Teste baseado em eventos  
(Capta os cenários de eventos sensíveis ao uso do dispositivo móvel).

**Figura 33.** Tela de selecionar abordagens de teste

A partir das estratégias de teste selecionadas, foi realizada outra busca informal na literatura técnica em busca de abordagens de teste que poderiam ser utilizadas no teste de aplicações móveis. A partir das abordagens resultantes, foi realizado um filtro selecionando as mais utilizadas atualmente no ambiente industrial. Vale ressaltar que podem existir outras abordagens de teste não mencionadas. Assim como no passo anterior mais de uma abordagem de teste pode ser selecionada.

### 3.4.1.3. Selecionar testes automatizados

Após a seleção de abordagens é questionado ao usuário se o mesmo deseja utilizar testes automatizados. Essa opção é responsável pela seleção ou não de ferramentas de apoio (Figura 34).

\* Serão utilizados testes automatizados?

Sim

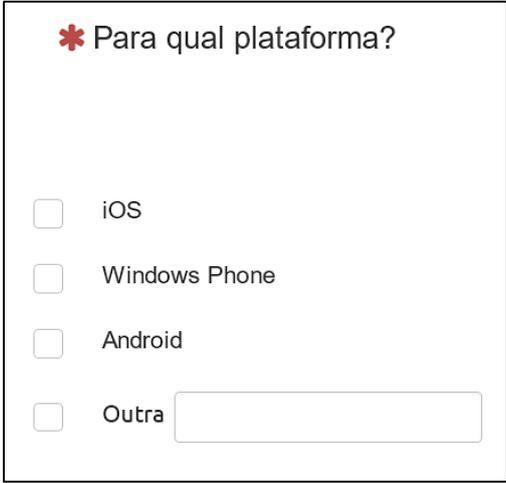
Não

**Figura 34.** Tela seleção testes automatizados

Caso deseja-se realizar testes automatizados, as ferramentas serão selecionadas de acordo com a estratégia de teste, abordagem, plataforma e ambiente de execução selecionados. Nessa pergunta apenas uma opção pode ser selecionada.

#### 3.4.1.4. Selecionar plataforma de teste

A seguinte pergunta só é acionada ao selecionar testes automatizados. Nesse momento, o usuário deve escolher entre as plataformas de teste mais conhecidas, sendo possível selecionar uma ou mais plataformas (Figura 35).



\* Para qual plataforma?

iOS

Windows Phone

Android

Outra

**Figura 35.** Plataformas de teste

As plataformas de execução incluídas na pergunta foram escolhidas após uma pesquisa de mercado buscando as que tinham maior fatia de mercado e percentual de usuários. A pergunta também consta com uma opção para inserção de outras plataformas, porém não há nenhuma ferramenta associada a essa opção.

#### 3.4.1.5. Selecionar ambiente de execução

Após selecionar o uso ou não de automação, deve-se escolher o ambiente onde os testes serão executados. A ferramenta *MTPLT* disponibiliza um conteúdo de ajuda caso ocorra alguma dúvida a respeito das definições (Figura 36).

Os ambientes de execução foram selecionados baseando-se na maturidade dos mesmos e utilização em testes de aplicações móveis. Atualmente, os testes em dispositivos e emuladores são os mais conhecidos e apresentam maior maturidade, porém a busca constante em inovação e redução de custos vem trazendo um crescimento constante nos testes realizados na nuvem e em larga escala. Todos os ambientes possuem estratégias e ferramentas específicas. Nessa questão pode ser selecionados mais de um ambiente de execução.

\* Qual o ambiente de execução a ser utilizado nos testes? (Para dúvidas a respeito das definições clique [aqui](#))

Dispositivo  
(Pode verificar funções específicas do dispositivo, comportamentos e requisitos não-funcionais).

Emulador  
(Faz uso de um emulador que cria uma versão da máquina virtual de um dispositivo móvel para estudo em um computador pessoal).

Nuvem (*Cloud Testing*)  
(Nuvem de dispositivos móveis conectados em uma rede privada que pode suportar serviços de testes em grande escala).

Larga Escala (*Crowd Testing*)  
(Uso de profissionais de testes voluntários, contratados ou uma comunidade de usuários finais, juntamente com uma infraestrutura de teste com um servidor de gerenciamento de serviços de apoio a diversos usuários).

**Figura 36.** Tela de seleção de ambiente de testes

### 3.4.1.6. Relatório

Como resultado da utilização da ferramenta é disponibilizado um relatório contendo o processo indicado para o teste da aplicação, o que deve ser realizado em cada atividade e indicação de ferramentas que podem apoiar a execução dos testes (Figura 37).

**Design & Implementação (Testes funcionais / comportamentais)**

O processo adotado para os testes funcionais / comportamentais é apresentado abaixo, sendo detalhadas as tarefas a serem realizadas em cada atividade.

```

graph LR
    1[1. Selecionar itens e características de teste] --> 2[2. Descrever estratégias de teste]
    2 --> 3[3. Identificar casos e procedimentos de teste]
    3 --> 4[4. Priorizar casos e procedimentos de teste]
    4 --> 5[5. Especificar critérios de aprovação / rejeição]
    5 --> 6[6. Gerar scripts de teste]
    6 --> 7[7. Documentar design & implementação de teste]
  
```

**1 - Selecionar itens e características de teste**

Durante a tarefa de Selecionar itens e características de teste (1) devem ser selecionados o tipo de teste especificado no Plano de Teste, saída da fase de planejamento de teste.

Ao escolher testes funcionais e comportamentais os seguintes itens e características são passíveis de teste:

**Figura 37.** Tela de relatório da aplicação MTPLT.

O relatório é fruto de todas as decisões realizadas pelo usuário do sistema. Um diagrama é apresentado simulando o processo e atividades a serem seguidos. As atividades são detalhadas de forma profunda e descritiva. O propósito do relatório é indicar passos, ferramentas e estratégias a serem utilizadas. Dessa forma, nem toda a informação apresentada deve ser completamente adotada, podendo ser adaptada de forma a melhor atender aos requisitos e processos da

organização. O relatório exibido abrange as fases de Design & Implementação e Execução, realizando uma varredura pontual e detalhada por cada atividade que compõe essas fases. No relatório inicialmente são visualizados todas as informações pertinentes a Design & Implementação de uma ou mais estratégias escolhidas e depois de Execução. A ferramenta foi pensada dessa forma para preservar a execução padrão de um processo de teste de forma ininterrupta.

Ao final da visualização encontra-se disponível um documento em PDF para download contendo o material do relatório e todas as decisões tomadas para alcançar aquele resultado, podendo ser utilizado para reproduções futuras. Também são disponibilizados *templates*, guias e exemplos a respeito de documentações a serem produzidas nessas fases.

No próximo capítulo, será apresentado um estudo que teve como objetivo avaliar a viabilidade da LPrS por meio da utilização da ferramenta *MTPLT* em relação a adequação ao mercado na instanciação de processos de teste para aplicações móveis.

## 4. AVALIAÇÃO DA LINHA DE PROCESSOS PROPOSTA

### 4.1. Definição do estudo de viabilidade

#### 4.1.1. Propósito

O propósito deste estudo foi responder à questão: “A utilização da ferramenta MTPL é viável analisando sua utilidade e facilidade de uso?”. Por utilidade de uso, entende-se neste estudo o grau em que o usuário acredita que o uso de um sistema particular pode melhorar o seu desempenho, e por facilidade percebida de uso, o grau em que uma pessoa acredita que o uso de um sistema de informação será livre de esforço.

#### 4.1.2. Perspectiva

A perspectiva é do ponto de vista dos especialistas. Nesse contexto, os especialistas mais próximos são os gerentes de teste/gerentes de qualidade e analistas de teste/líder técnico de testes que desejam analisar a utilidade e facilidade de uso do MTPL na atividade de teste de aplicações móveis.

#### 4.1.3. Objetivo específicos

- **Analisar** a ferramenta de apoio à definição de processos de teste de software em aplicações móveis (MTPL);
- **Com o propósito** de caracterizá-lo;
- **Com relação** à utilidade e facilidade de uso;
- **Do ponto de vista** dos especialistas;
- **No contexto** de teste de aplicações móveis.

#### 4.1.4. Questões e métricas

- **Q1)** Qual é a utilidade da ferramenta MTPL no que diz respeito à percepção dos especialistas?
  - **Métrica:** número de participantes favoráveis ao uso da ferramenta proposto em função de sua utilidade.
- **Q2)** A ferramenta apresentado é de fácil utilização no ambiente organizacional no que

diz respeito à percepção dos especialistas?

- **Métrica:** número de participantes que concluíram as tarefas e responderam as questões de avaliação de facilidade percebida de uso.

## **4.2. Planejamento do estudo**

Este estudo tem como objetivo a avaliação da ferramenta MTPLT, que permite a definição de processos de teste para aplicações móveis, no qual foram utilizados conceitos de sistematização de processos. Com os resultados obtidos a partir da realização desse estudo é possível caracterizar a ferramenta proposto e o corpo de conhecimento decorrente de sua condução fornecerá informações que permitam evoluir a ferramenta.

Para avaliação, foi escolhido o TAM (Modelo de Aceitação de Tecnologia) (DAVIS et al., 1989).. Como o modelo é comportamental, só pode referir-se às questões diretamente relacionadas com o usuário e suas percepções sobre o uso do sistema. Por isso, os *constructos* devem ser desenvolvidos de modo a captar opiniões pessoais e tratar suposições a respeito de terceiros (pessoas ou instituições) (SALEH, 2004). Este modelo é útil não só para prever, mas também para descrever, de forma que pesquisadores e profissionais possam identificar o porquê da não aceitação de um sistema ou tecnologia em particular pelos usuários e, conseqüentemente, implementar os passos corretivos adequados (DAVIS et al., 1989).

A partir da fundamentação teórica fornecida pelo modelo TAM, foram identificadas duas dimensões principais: (1) utilidade percebida de uso e (2) facilidade percebida de uso. Para cada variável, foi construída uma escala de Likert composta por 5 cinco itens, variando entre 1 (um) e 5 (cinco) pontos. O grau de concordância cresce conforme maior for o número de pontos assinalados. O ponto 3 (ponto central) da escala foi considerado como neutro.

É importante ressaltar que o objeto do instrumento de pesquisa - questionário - foi o indivíduo. Portanto, as escalas foram preparadas em primeira pessoa com intuito de facilitar a interpretação dos itens pelos respondentes.

### **4.2.1. Seleção dos participantes**

Os especialistas participantes deste estudo devem possuir experiência no ambiente de teste de aplicações móveis. Inicialmente, eles respondem a um questionário de caracterização de perfil técnico. Os participantes foram selecionados a partir de uma busca na rede social de profissionais *LinkedIn* e contato direto com empresas de software por meio de ex-alunos da universidade.

Ao responder o questionário de caracterização, o especialista pode assumir os seguintes

papéis:

- Líder técnico de teste / Analista de teste
- Gerente de teste / Gerente de qualidade

#### 4.2.2. Seleção de grupos

Quando possível, os participantes do estudo serão divididos em dois grupos com o mesmo número de especialistas. Caso haja número ímpar, o especialista com experiência mais distante dos demais será descartado. A distribuição dos participantes nos grupos é realizada de forma aleatória. Cada grupo possui uma atividade específica a ser realizada durante a utilização da ferramenta, conforme descrito abaixo:

- **Grupo A:** Como participante do Grupo A, utilizando a ferramenta deve-se tentar gerar um processo de teste para uma aplicação que possua características **funcionais**, utilizando abordagem de **teste de gestos**, de forma **não-automatizada** para qualquer ambiente e plataforma de execução.
- **Grupo B:** Como participante do Grupo B, utilizando a ferramenta deve-se tentar gerar um processo de teste para aplicação que possua características de **usabilidade não-empíricas**, utilizando a abordagem de **lista de verificação** de forma **automatizada** para qualquer plataforma e ambiente de execução.

#### 4.3. Execução do estudo

O estudo de viabilidade foi conduzido de forma *offline*, ou seja, os participantes receberam um e-mail informando a atividade específica a ser realizada na ferramenta, um link para utilização da ferramenta e outro para responder ao questionário e poderiam responder no momento que julgarem adequados.

Ao acessar o questionário, os especialistas preencheram um formulário de caracterização, visando identificar o perfil e experiência em testes de aplicações móveis, e concordaram com o Termo de Consentimento Livre e Esclarecido (TCLE).

O estudo consiste na utilização da ferramenta, tentativa de realizar a atividade específica do grupo e preenchimento do questionário de avaliação da ferramenta, não sendo necessário qualquer treinamento prévio para utilização da ferramenta MTPLT disponível no link: [goo.gl/Or3vwD](http://goo.gl/Or3vwD).

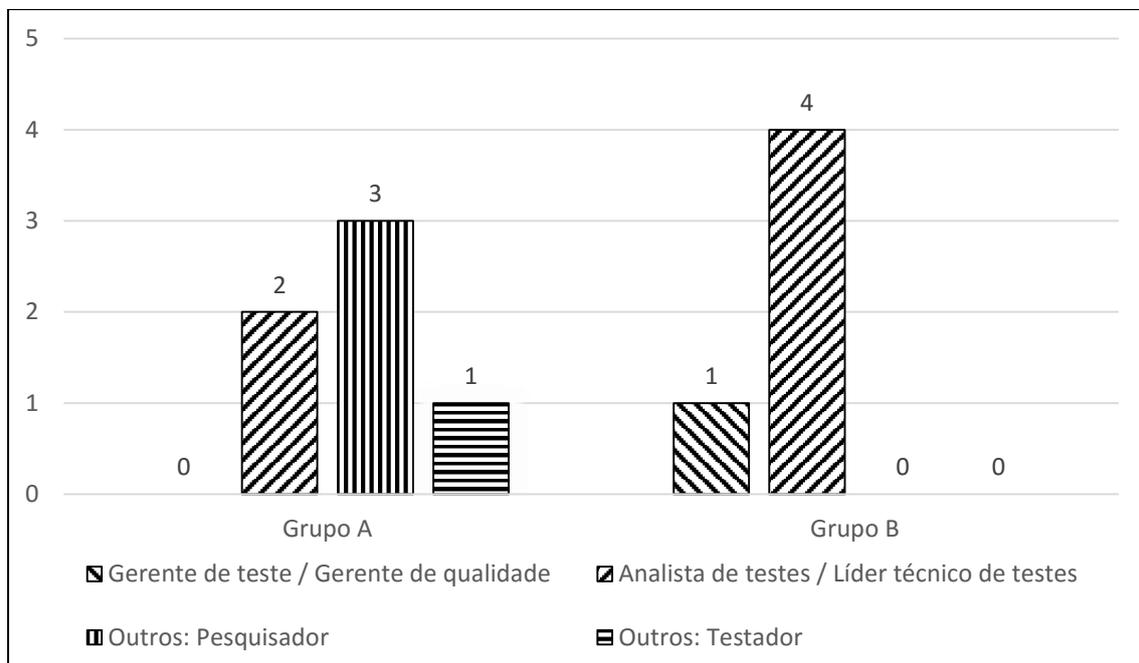
Foram convidados cerca de 40 participantes, divididos em dois grupos de forma aleatória,

sendo obtidas 10 respostas, 5 de cada grupo.

## 4.4. Resultados do estudo

### 4.4.1. Caracterização dos participantes

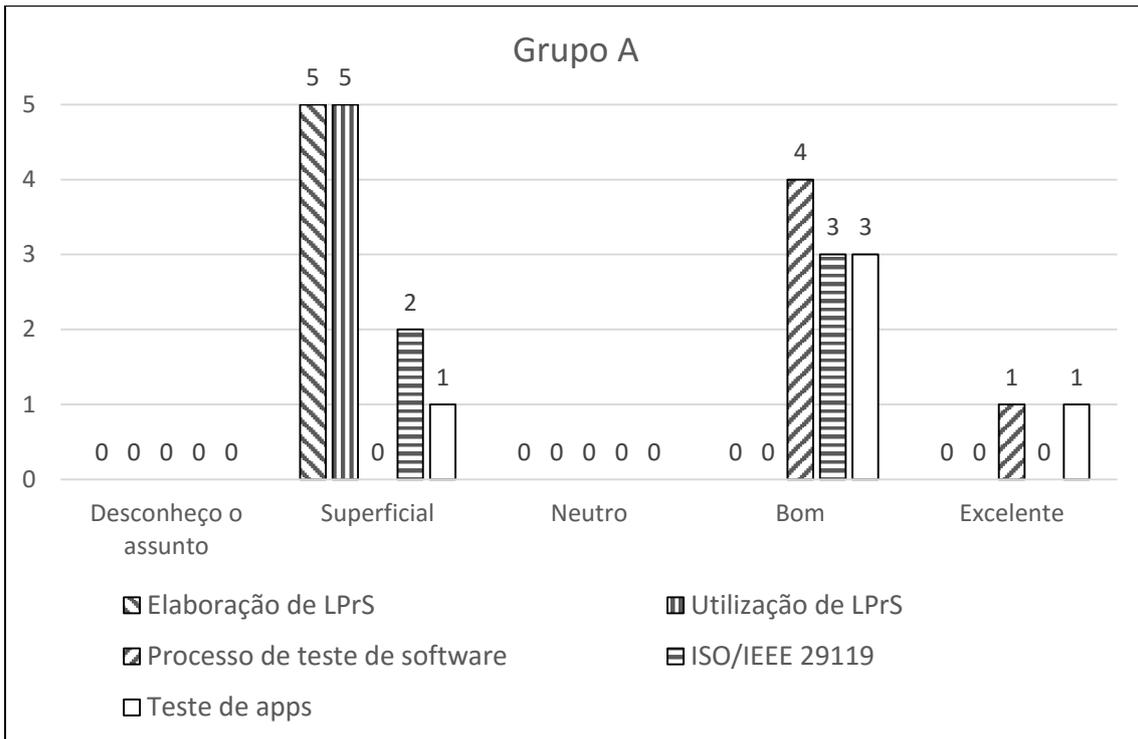
Ao iniciar o questionário de avaliação, era necessário preencher um formulário de caracterização, com questões a respeito de experiência, papel desempenhado e conhecimento prévio a respeito de alguns assuntos. Em todos os quesitos foi mantida a divisão em dois grupos, pois cada um possuía uma atividade específica a ser realizada. Os resultados no quesito papéis desempenhados podem ser observados na Figura 38.



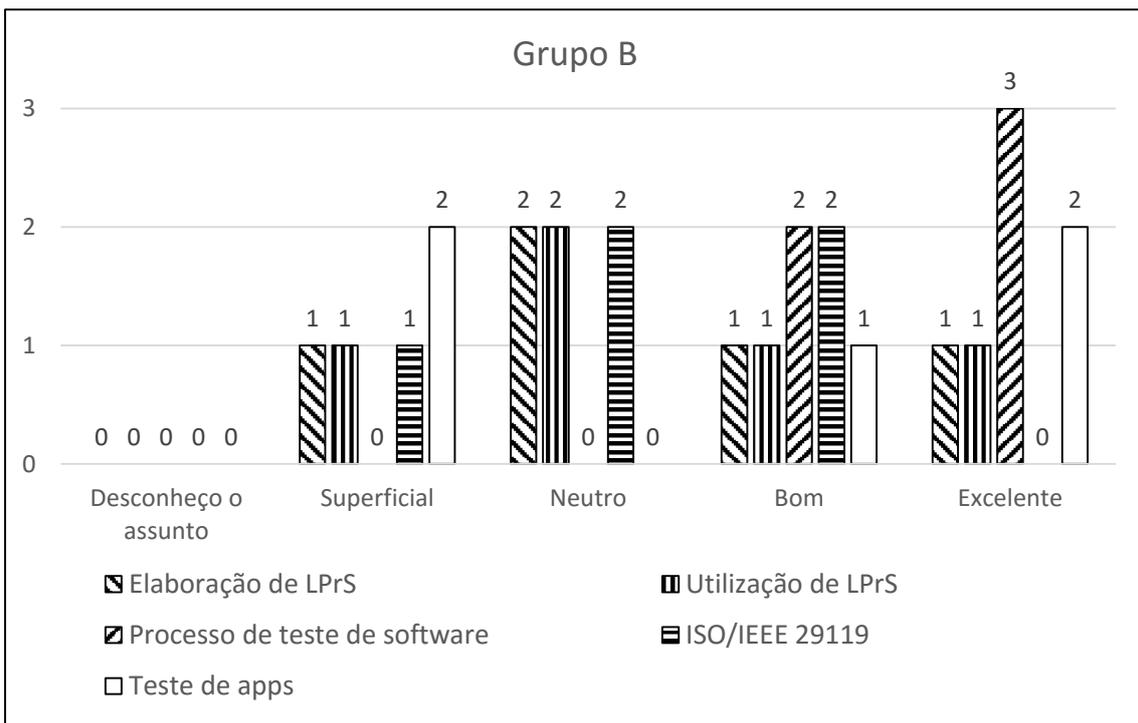
**Figura 38.** Papel desempenhado por participantes do experimento

A Figura 38 demonstra que no Grupo A um especialista se classificou como testador e três como pesquisadores na área de teste de software. Os outros dois participantes afirmaram que eram Analistas de testes ou Líderes técnicos de testes. No Grupo B pode ser observado outro cenário, onde a maior parte dos especialistas era Analista de testes/Líder técnico de testes e apenas um se considerou como Gerente de testes/Gerente de qualidade.

Após a análise de papéis desempenhados, os especialistas foram questionados a respeito de conhecimento prévio em alguns quesitos. As respostas foram classificadas utilizando a escala de *Likert* com 5 fatores (Desconheço o assunto, superficial, bom, excelente e neutro). A Figura 39 apresenta os resultados para o Grupo A e a Figura 40 para o Grupo B.



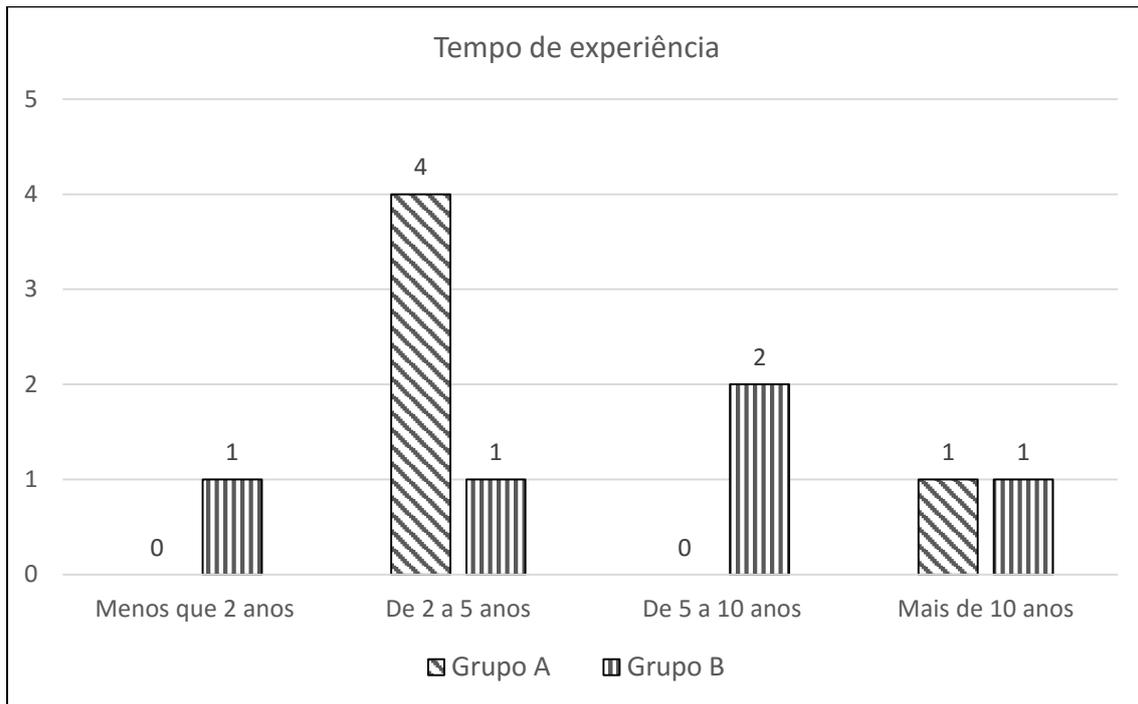
**Figura 39.** Conhecimentos prévios do Grupo A.



**Figura 40.** Conhecimentos prévios do Grupo B.

Observando as figuras 39 e 40 podemos perceber que nenhum dos participantes desconhecia qualquer um dos assuntos, tornando assim todos aptos a participarem do experimento. No próximo gráfico (Figura 41), pode-se observar o tempo de experiência

em teste de aplicações móveis dos especialistas.



**Figura 41.** Tempo de experiência em teste de aplicações móveis

Analisando o gráfico um participante de cada grupo possuía mais de 10 anos de experiência e os demais estavam distribuídos nas demais classificações, sendo também apenas um participante do Grupo B com menos de 2 anos de experiência.

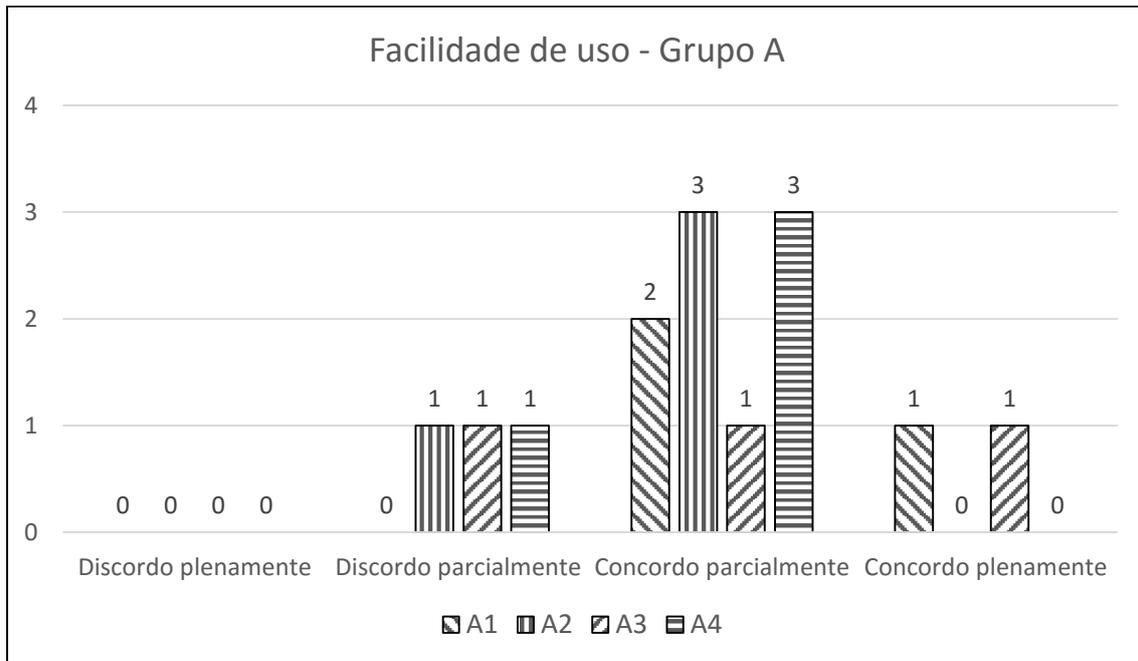
#### 4.4.2. Análise quantitativa

Nesta seção, foi feita análise de forma qualitativa da opinião dos participantes em relação à utilização da ferramenta e execução da atividade específica. O questionário aplicado tem como objetivo avaliar a facilidade de uso (4 questões) e a utilidade percebida (6 questões). Também foi utilizada a escala *Likert* com 5 fatores para opções de respostas (Discordo plenamente, discordo parcialmente, neutro, concordo parcialmente e concordo plenamente). Como o intuito é avaliar o grau de concordância ou discordância, respostas classificadas como neutro não serão consideradas.

##### 4.4.2.1. Facilidade de uso

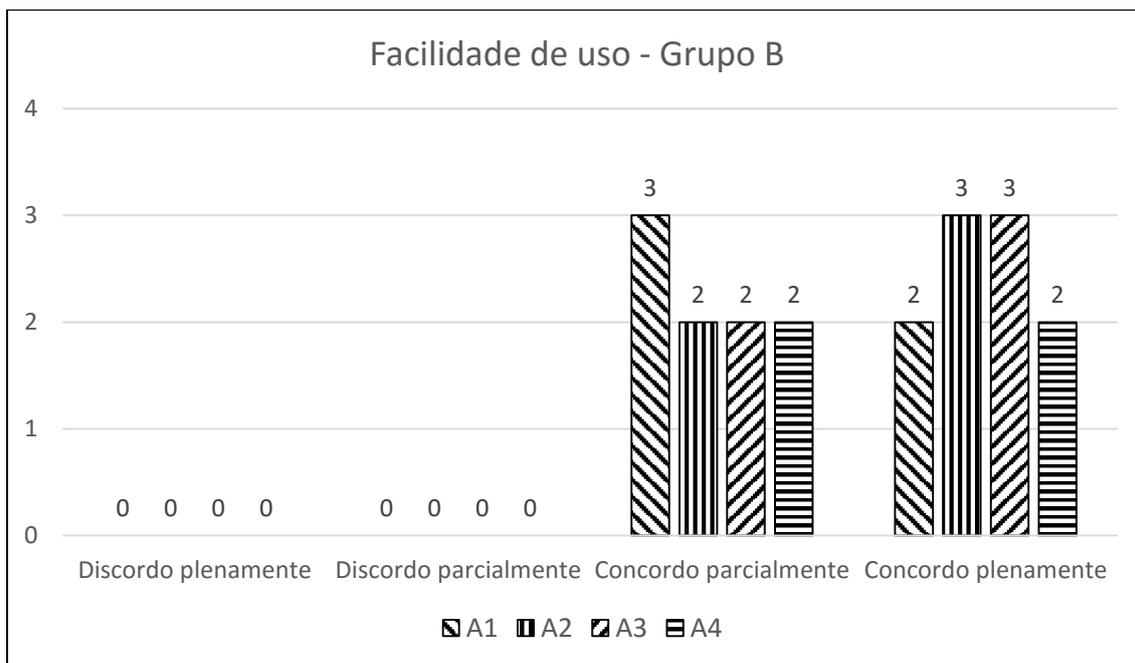
A facilidade de uso tem o objetivo de averiguar se a ferramenta foi bem projetada e apresenta uma linguagem clara e compreensível. Quatro afirmações foram utilizadas para avaliar a facilidade de uso, são elas:

- A1) Realização da atividade específica (dependente de cada grupo);
- A2) Aprender a utilizar o MTPLT foi fácil;
- A3) O acesso ao MTPLT é simples;
- A4) Minha interação com o MTPLT é clara e compreensível.



**Figura 42.** Facilidade de uso do Grupo A

Analisando a Figura 42 pode-se perceber que grande parte dos especialistas do Grupo A apresentaram um grau de concordância com a facilidade de uso da ferramenta desenvolvida. Houve apenas um especialista que apresentou discordância em 3 questões.



**Figura 43.** Facilidade de uso do Grupo B.

Analisando a facilidade de uso do Grupo B (Figura 43), observou-se um total nível de concordância dos especialistas. Dessa forma pode-se aferir que a ferramenta é de fácil utilização.

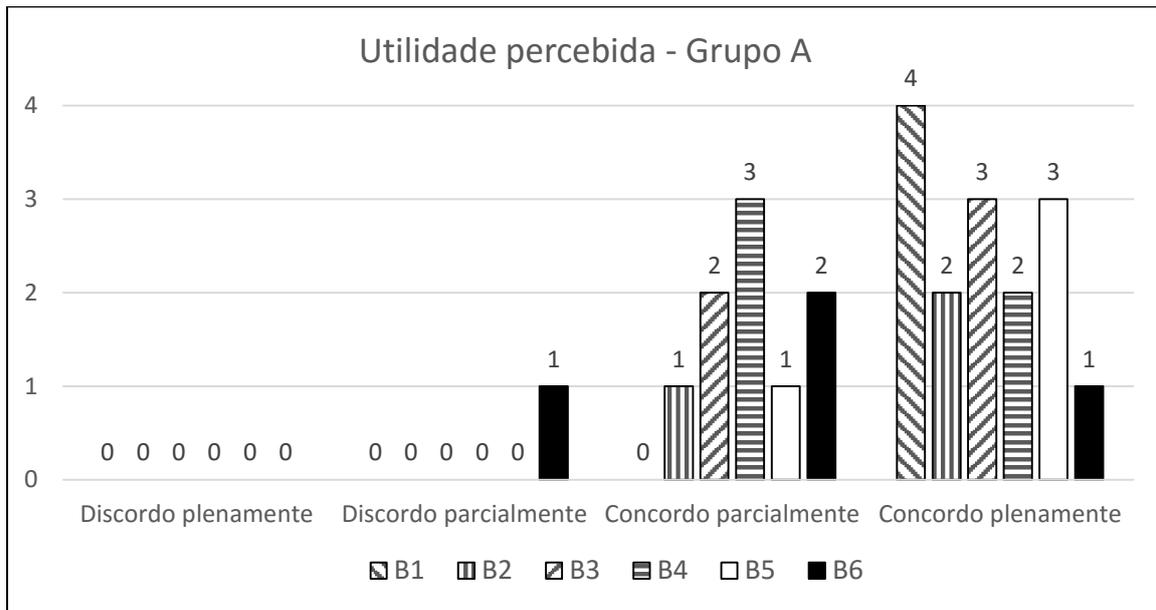
#### 4.4.2.2. Utilidade percebida

Utilidade percebida pode ser definida como o quanto os especialistas acreditam que a ferramenta está alinhada ao mercado, objetivo e o quão útil é ao público-alvo definido. Foram seis afirmações que auxiliam a avaliar esse aspecto.

- B1) Utilizar o MTPLT é importante e adiciona valor ao meu trabalho;
- B2) Eu gostaria de trabalhar com o MTPLT;
- B3) O MTPLT é útil na derivação de processos para teste de aplicações móveis;
- B4) Usar o MTPLT produz os resultados que espero de uma sistematização de processos;
- B5) Usar o MTPLT minimizou o tempo de definição de processos de teste e ferramentas a serem utilizadas;
- B6) Compreendo claramente as informações apresentadas no relatório provido pelo MTPLT.

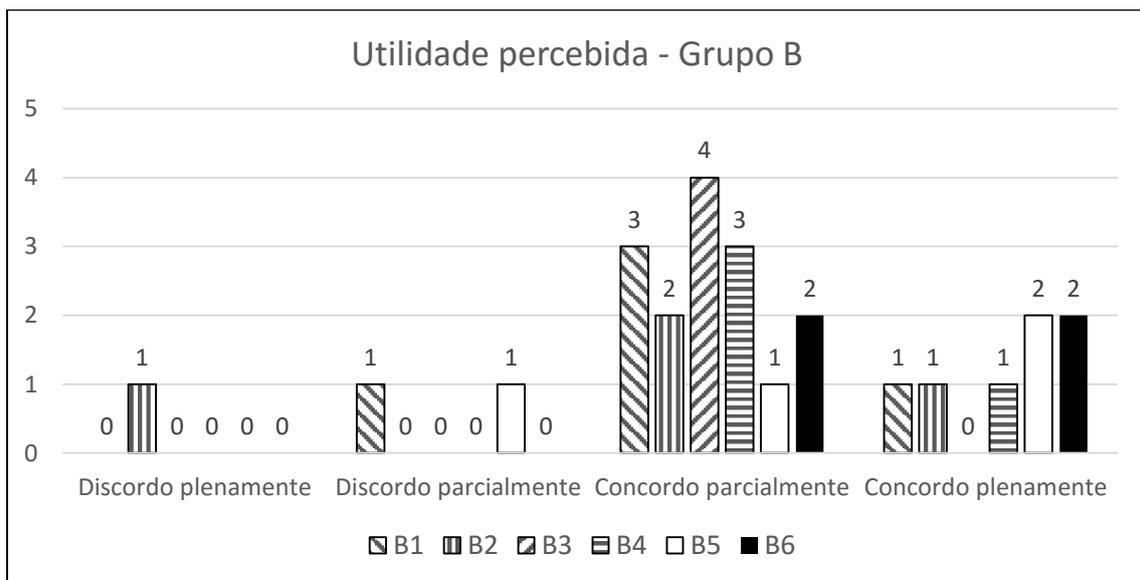
Na Figura 44 pode ser observada a utilidade percebida avaliada pelos especialistas do

Grupo A.



**Figura 44.** Utilidade percebida do Grupo A

Observando os resultados da Figura 44 pode-se aferir que a maior parte dos especialistas se mostrou favorável a utilidade percebida da ferramenta, sendo resultante apenas um ponto negativo na afirmação a respeito da compreensão do relatório, porém ao investigar a discordância foi informado que havia um problema quanto ao cache da aplicação e as imagens não estavam sendo apresentadas de forma devida, logo isso causou uma percepção negativa por parte do especialista. A seguir serão analisados os dados do Grupo B na Figura 45.



**Figura 45.** Utilidade percebida do Grupo B.

A Figura 45 mostra que houve um grande grau de concordância entre os participantes do experimento, porém foram obtidas três discordâncias nas afirmações B1 – Utilizar o MTPLT é importante e adiciona valor ao meu trabalho, B2 – Eu gostaria de trabalhar com o MTPLT e B5 – Usar o MTPLT minimizou o tempo de definição de processos de teste e ferramentas a serem utilizadas, ao analisar o perfil dos especialistas que assinalaram essas respostas foi extraído que todas pertenciam ao mesmo participante e seu perfil era de Pesquisador na área de teste, logo essa pode ser uma possível causa da ferramenta não apresentar uma utilidade direta ao participante.

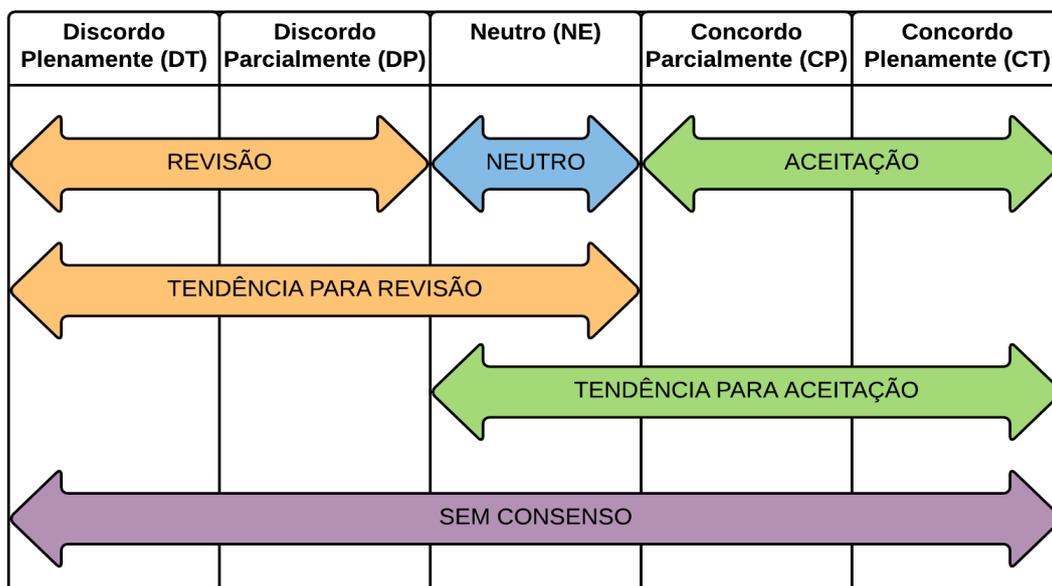
#### **4.4.2.3. Tendências**

Com base na agregação de resultados, é possível identificar, para cada fator da linha de processo proposta (Utilidade percebida e Facilidade de uso), se é possível considerar o item como satisfatório ou se é necessário introduzir alguma mudança. Apenas nessa seção serão consideradas as respostas assinaladas como neutras pelos especialistas. Para essa análise optou-se por uma escala, apresentada na Figura 46.

- Aceito: caso todos os especialistas assinalem os itens CP (concordo parcialmente) ou CT (concordo plenamente), o item será considerado como adequado.
- Tendência de aceitação: caso haja algum especialista apontando para o item NE (neutro), então o item será considerado como tendo uma tendência de aceitação, mas as sugestões serão analisadas. Será considerado nesta situação também o caso em que houver menos do que 3 votos para o lado oposto.
- Neutro: caso todos os especialistas apontem para o item NE (neutro), então será considerado que são indiferentes. Neste caso o item também será considerado como aceito.
- Revisão: caso todos os especialistas assinalem os itens DT (discordo plenamente) ou DP (discordo parcialmente), o item será considerado como sujeito à revisão.
- Tendência de rejeição: caso haja algum especialista apontando para o item NE (neutro), então o item será considerado como tendo uma tendência de rejeição, mas as sugestões serão analisadas. Será considerado nesta situação também o caso em que houver menos do que 3 votos para o lado oposto.
- Demais casos: quando as opiniões estiverem distribuídas entre as várias faixas, o item será considerado como sem consenso. Neste caso serão avaliadas as sugestões e

incorporadas, caso pertinente.

A Figura 46 resume a abordagem adotada para análise dos resultados de tendências.



**Figura 46.** Interpretação dos resultados.

Sendo assim, as respostas dos especialistas foram sumarizadas a fim de identificar as tendências dos dois fatores apresentados anteriormente. Foram considerados os 10 especialistas que responderam ao questionário, sendo que foram criadas 4 questões para a facilidade de uso e 6 questões para a utilidade percebida. Sendo assim o total de pontos para a facilidade de uso é 40 e para a utilidade 60. A sumarização pode ser observada nas Tabelas 31 e 32.

**Tabela 35.** Tendência da utilidade percebida

Utilidade percebida						
Questão	DT	DP	NE	CP	CT	Análise
Utilizar o MTPLT é importante e adiciona valor ao meu trabalho.	0	1	1	3	5	Tendência para aceitação
Eu gostaria de trabalhar com o MTPLT.	1	0	3	3	3	Tendência para aceitação
O MTPLT é útil na derivação de processos para teste de aplicações	0	0	1	6	3	Tendência para aceitação

móveis.						
Usar o MTPLT produz os resultados que espero de uma sistematização de processos.	0	0	1	6	3	Tendência para aceitação
Usar o MTPLT minimizou o tempo de definição de processos de teste e ferramentas a serem utilizadas;	0	1	2	2	5	Tendência para aceitação
Compreendo claramente as informações apresentadas no relatório provido pelo MTPLT.	0	1	2	4	3	Tendência para aceitação
Total	1	3	10	24	22	60

**Tabela 36.** Tendência da facilidade de uso

<b>Facilidade de uso</b>						
Questão	DT	DP	NE	CP	CT	Análise
Realizar a atividade específica foi fácil.	0	0	2	5	3	Tendência para aceitação
Aprender a utilizar o MTPLT foi fácil.	0	1	1	5	3	Tendência para aceitação
O acesso ao MTPLT é simples.	0	1	2	3	4	Tendência para aceitação
Minha interação com o MTPLT é clara e compreensível.	0	1	2	5	2	Tendência para aceitação
Total	0	3	7	18	12	40

A partir das tabelas 31 e 32 foram identificados pontos que deveriam ser analisados, a análise é apresentada na próxima seção por meio dos comentários dos especialistas.

### 4.4.3. Análise qualitativa

O questionário apresentava um campo para comentários e sugestões a respeito da utilização da ferramenta, de um total de 10 participantes apenas 6 responderam ao campo. Dessa forma as respostas foram analisadas de acordo com o perfil do participante (Tabelas 33 a 38).

**Tabela 37.** Sumário do participante 1

Participante 1	
Experiência	Papel
Mais de 10 anos	Pesquisador na área de teste de software
Creio que o principal problema é que não ficou claro inicialmente qual era a ideia da avaliação. Depois de um tempo percebi que já estava usando a ferramenta e não somente o conceito. Acho que seria importante declarar no início o que a ferramenta oferece ao usuário.	

O participante 1 não compreendeu o objetivo da ferramenta, indicando que deveria ser mostrado no início da utilização. Sendo assim foi analisado o conteúdo introdutória da ferramenta, que constituía em:

A ferramenta *Mobile Testing Process Line Tool* tem como objetivo guiar o usuário na definição do processo a ser adotado no teste de aplicações móveis, sugerindo ferramentas, atividades, *templates*, tipos e técnicas de teste a serem utilizadas. A ferramenta é focada nas fases técnicas do teste de software (Design & Implementação e Execução), podendo ainda apresentar conceitos gerais a respeito das demais fases. Como passo inicial é necessário ter conhecimento dos seguintes aspectos do teste:

- Itens e características a serem testadas;
- Plataformas a serem utilizadas;
- Ambientes de execução;
- Se necessária automatização de testes.

Dessa forma após uma análise em pares não foi identificado nenhum problema a respeito do texto apresentado, o que pode ter ocorrido é que o participante pode ter passado despercebido

pelo conteúdo inicial da ferramenta.

**Tabela 38.** Sumário do participante 2

<b>Participante 2</b>	
<b>Experiência</b>	<b>Papel</b>
De 2 a 5 anos	Pesquisador na área de teste de software
Utilizando o browser Google Chrome na plataforma Windows não foi possível visualizar as figuras apresentados nos resultados gerados pela ferramenta	

Os comentários do participante 2 são similares ao item 1 do participante 3, após o feedback foi analisada a ferramenta em busca de problemas que estivessem impedindo a visualização das imagens, e foi identificado que o servidor onde a ferramenta estava hospedada apresentava um problema no cache que fazia com que após uma queda de energia (resultando no desligamento do servidor), as imagens armazenadas eram excluídas. Sendo necessária uma nova adição de todas as imagens ao servidor. O problema foi solucionado temporariamente para continuidade do experimento e após sua conclusão será buscada uma solução definitiva.

**Tabela 39.** Sumário do participante 3

<b>Participante 3</b>	
<b>Experiência</b>	<b>Papel</b>
De 2 a 5 anos	Analista de testes / Líder técnico de testes
Algumas sugestões são: 1 - Adicionar Fluxos, imagens, gráficos que possam ajudar na compreensão de alguns processos (desconsidere caso já exista, é que as imagens contidas no sistema não foram carregadas). 2 - Outra Sugestão seria separar a parte do relatório das funcionalidades que o antecedem, tipo ao gerar o Relatório apresentar um formato diferente. Sobre o conteúdo achei excelente, rico e válido, a disposição das informações também está bacana, no entanto, pode melhorar mais ainda. Sobre a ferramenta em si gostei da proposta e acredito que ela auxilia bastante nas definições propostas.	

O comentário 1 do participante já foi respondido anteriormente, a respeito do comentário 2 quanto a separação isso já ocorre, porém para âmbito de avaliação a ferramenta MTPLT e o questionário de avaliação foram integrados para facilitar a resposta do especialista, tendo em vista

que anteriormente o especialista deveria utilizar dois links de acesso o que poderia gerar certa confusão. Por esse motivo o especialista considerou que tudo fosse integrado o que causaria certa confusão ao responder. Portanto a sugestão já apresentava solucionada em outra instância da ferramenta.

**Tabela 40.** Sumário do participante 4

<b>Participante 4</b>	
<b>Experiência</b>	<b>Papel</b>
De 2 a 5 anos	Testador
Olá, Poderia ter um menu de navegação no canto informando "em que passo estou" pra evitar ficar clicando no anterior e próximo várias vezes ou até mesmo pra ter noção do que eu fiz e do que ainda falta. Quanto ao procedimento, está bem fácil de ser usado só fiquei meio confuso no início quando a primeira pergunta já veio bem direta. Acredito que a intenção é ser o mais sucinta e eficiente possível. Porém, pode ser que precise de uma contextualização da etapa apenas pra ficar mais amigável. Coisa simples mesmo como por exemplo: Em seu processo de teste na fase de xxx você irá utilizar as estratégias yyy. No mais, é um excelente trabalho e com certeza e eu usaria, mas isso já foi dito no questionário. Parabéns pelo trabalho.	

O comentário do participante 4 apresentou duas sugestões: quanto ao uso de um menu informando o que já foi concluído e uma mudança no texto da ferramenta inserindo as questões em um contexto. Ambas as mudanças foram consideradas coerentes e acatadas, sendo integradas em uma nova versão da ferramenta.

**Tabela 41.** Sumário do participante 5

<b>Participante 5</b>	
<b>Experiência</b>	<b>Papel</b>
De 5 a 10 anos	Analista de testes / Líder técnico de testes
No cenário existente hoje, acho que dificilmente a ferramenta será utilizada, ou caso utilizada, dificilmente o processo como está sugerido será seguido devido ao tempo, principalmente se tratando de aplicações mobile, onde os prazos são ainda mais curtos. Acredito que ela serve mais como um direcionamento para alguém que não tenha ideia de como definir um processo de testes. Para o caso em que vários tipos de teste serão realizados para a mesma aplicação, ao invés de sugerir um processo para cada tipo de teste, sugiro agrupar as atividades repetidas e montar UM só processo, onde as diferenças podem ser destacadas apenas nas atividades	

divergentes indicando para que tipo de teste um determinado trecho da atividade se aplica.

O participante 5 apresentou algumas críticas a respeito do uso e conteúdo da ferramenta, ao seu ver a ferramenta não seria adequada a organizações que adotassem a metodologia ágil na etapa de desenvolvimento e teste de software, o que foi considerado, pois o processo adaptado na ferramenta era voltado a metodologia tradicional de teste de software. Sendo assim em uma versão futura poderão ser incluídos processos que possam ser aplicados na metodologia ágil.

Outra consideração do participante foi quanto a quem seria o público alvo da ferramenta, a consideração foi considerada importante, pois ao definir a ferramenta o público alvo inicial seriam gerentes de teste / gerentes de QA, porém após o levantamento do especialista foi percebido que a ferramenta poderia também ser aplicada a pessoas com pouca experiência, pois apresenta um passo a passo detalhado a respeito das atividades a serem executadas no processo.

Como última sugestão têm-se a integração dos processos apresentados para o teste de uma mesma aplicação, evidenciando apenas o que deveria ser diferente para cada tipo de teste específico, tendo em vista que a maior parte deles possuem as mesmas atividades diferenciando apen. A sugestão foi considerada válida, porém por limitações da ferramenta não pode ser integrada.

**Tabela 42.** Sumário do participante 6

<b>Participante 6</b>	
<b>Experiência</b>	<b>Papel</b>
Mais de 10 anos	Gerente de testes / Gerente de qualidade
Gostei do relatório de resultado, pois é bem detalhado e identifica ferramentas para mobile testing. No entanto, senti falta da informação sobre a licença das ferramentas, se são gratuitas ou pagas. Na seção de ferramentas achei que o texto foi copiado dos fornecedores e alguns textos não ficaram adequados para o relatório, p.e. "Como os especialistas em testes automatizados, estamos de acordo com..". Como sugestão, se houver alguma referência para os Critérios de Parada, seria interessante.	

Analisando os comentários do participante 6 percebeu-se uma maior preocupação em relação a ferramentas de automatização, o que considerando seu perfil de gerente de testes / gerente de qualidade evidencia que dependendo do perfil do especialista a ferramenta tem uma aplicação e foco diferente. Seus comentários a respeito do texto e licenciamento das ferramentas foi acatado e será incluído em uma nova versão da ferramenta a ser disponibilizada.

## 4.5. Ameaças à validade

### 4.5.1. Validade interna

- **Instrumentação:** os instrumentos que foram utilizados (formulários online) passaram por revisão e foram submetidos a um estudo piloto que apontou melhorias implementadas;
- **Seleção:** como mencionado anteriormente, para seleção dos participantes levou-se em consideração sua caracterização resultante do formulário de caracterização no questionário. Os grupos foram distribuídos de maneira igualitária de forma aleatória, com base na ordem de contato;

### 4.5.2. Validade externa

- **Participantes:** os participantes selecionados refletem o comportamento da população de gerentes de teste/gerentes de qualidade e analista de teste/líder técnico de teste, que de forma geral adquirem conhecimento sobre gerenciamento de testes e derivação de processos.

### 4.5.3. Validade de construção

- **Projeto do experimento:** Como descrito nos Capítulos 2 e 3, a ferramenta MTPLT passou por uma fase de concepção, baseado num corpo de conhecimento construído por meio de um mapeamento sistemático e sob uma prova de conceito para validar os conceitos propostos.
- **Fatores humanos (ou sociais):** os participantes não foram informados sobre o objetivo do experimento para não serem influenciados, participando assim de forma voluntária como uma maneira de adquirir conhecimento.

### 4.5.4. Validade de conclusão

- **Confiabilidade na Implementação do Tratamento:** Foi disponibilizado no e-mail um passo-a-passo das atividades a serem executadas, que foram submetidos a avaliação no teste piloto.

## 4.6. Considerações finais

Com este estudo foi possível analisar se a ferramenta MTPLT era fácil de usar e útil segundo a opinião de especialistas. Para isso, foi realizado um questionário com especialistas

utilizando o modelo de aceitação de tecnologia (TAM) como base. A ferramenta conseguiu resultados expressivos além de mostrar novas perspectivas aos especialistas. Foram utilizados dois fatores de avaliação Utilidade percebida e Facilidade de uso.

Em relação a facilidade de uso não houve nenhuma discordância total de qualquer elemento apresentado, a partir disso foi identificada uma tendência a aceitação da ferramenta relacionado a esse fator. Foram sugeridas algumas melhorias quanto a navegação e disposição de elementos da aplicação.

Analisando a utilidade percebida houve apenas uma discordância total, após averiguação verificou-se que a ferramenta não possuía um impacto direto no especialista que assinalou essa resposta, dessa forma a utilidade percebida poderia apresentar um viés. Foram sugeridas melhorias quanto a processos e adição de novas *features*.

Sendo assim, no próximo capítulo serão apresentadas as conclusões e trabalhos futuros.

## 5. CONCLUSÕES E TRABALHOS FUTUROS

### 5.1. Considerações finais

A atividade de teste de aplicações móveis tem se tornado uma tarefa árdua para os testadores e desenvolvedores, muitas vezes por não possuir um processo bem definido. A definição de um processo de teste torna-se complicado devido a grande variedade de ferramentas, tipos de teste e estratégias, tendo em vista que ao definir um processo de teste de aplicações móveis esses elementos devem estar bem definidos e alinhados, o gerente de teste pode ter dificuldades para definir esses padrões. Dessa forma todo o projeto é afetado, já que cada integrante utiliza metodologias diferentes, ocasionando uma falta de padronização e dificuldade de gerenciamento.

Como forma de oferecer suporte a esses desafios, esta pesquisa propôs uma Linha de Processo de Software para teste de Aplicações Móveis. A LPrS proposta tem o objetivo de derivar processos de teste de software para aplicações móveis. Foram selecionados métodos, estratégias, técnicas e ferramentas de teste que poderiam ser aplicados a aplicações móveis. A LPrS abrange quatro fases do processo de teste, sendo: Planejamento, Design & Implementação, Execução e Análise e Encerramento. Dentre as quais foram analisadas com maior ênfase as fases de Design & Implementação e Execução, por serem consideradas fases técnicas componentes do escopo do trabalho. Devido a grande quantidade de artefatos e buscando facilitar sua utilização e gerenciamento, foi proposta uma ferramenta para derivação dos processos intitulada de MTPLT (*Mobile Testing Process Line Tool*).

Para a identificação da estrutura da LPrS proposta foi realizada uma busca na literatura técnica objetivando uma definição concisa a respeito de linhas de processo de software, onde foi encontrado um mapeamento que consistia até meados de 2003. Por esse motivo foi necessária a realização de uma extensão do mapeamento buscando uma base de conhecimento atualizada e consistente com o cenário. Como segundo passo foi realizado um mapeamento buscando processos de teste de software, inicialmente o escopo definido era aplicados a teste de aplicações móveis, porém devido a quantidade de trabalhos retornados ser mínima, o escopo foi generalizado para teste de aplicações em geral.

Em seguida foi elaborada uma LPrS para definição de processos de teste quando aplicados a softwares móveis. A LPrS é composta dos principais conceitos de teste de software direcionados a aplicações móveis, tendo como objetivo a instanciação de um processo bem definido com respectivos papéis e atividades.

Como forma de apoiar a LPrS elaborada foi desenvolvida uma ferramenta Web

denominada MTPLT (*Mobile Testing Process Line Tool*). A ferramenta em si tem o objetivo de apoiar a utilização da LPrS, baseando-se em um modelo de questionário, possibilitando assim ao usuário sem conhecimento de LPrS sua utilização.

Como forma de avaliar a ferramenta proposta foi realizado um *survey* com especialistas com o objetivo de analisar a facilidade de uso e utilidade percebida. O *survey* proposto era composto de 17 questões, sendo 7 de caracterização, 4 com o objetivo de avaliar a facilidade de uso e 6 para avaliar a utilidade percebida. Após análise das questões a respeito da caracterização foi possível traçar um perfil dos participantes do *survey*, levando em consideração os quesitos: tempo de experiência, papel desempenhado e conhecimentos a respeito de aspectos pertinentes ao trabalho.

Em seguida os participantes eram direcionados a avaliação de facilidade de uso e utilidade percebida. Após análise dos 10 participantes foi identificada uma tendência a aceitação para os dois critérios escolhidos de avaliação. No quesito utilidade percebida de um total de 60 pontos foram obtidos: 1 – Discordo totalmente, 3 – Discordo parcialmente, 10 – Neutro, 24 – Concordo parcialmente e 22 – Concordo totalmente. Em facilidade de uso de um total de 40 pontos foram obtidos: 0 – Discordo totalmente, 3 – Discordo parcialmente, 7 – Neutro, 18 – Concordo parcialmente e 12 – Concordo totalmente. Sendo assim pode-se concluir que houve uma tendência a aceitação nos dois quesitos avaliados.

O *survey* aplicado possuía ainda um campo de comentários e sugestões de preenchimento opcional, de um total de 10 participantes apenas 6 contribuíram nesse campo, os principais pontos foram: layout da aplicação, adequação a metodologias ágeis, incremento de funcionalidades e problemas de compatibilidade. Os comentários foram analisados e implementados quando pertinentes, outros foram classificados como modificações futuras a ferramenta.

## **5.2. Contribuições da pesquisa**

As principais contribuições desta pesquisa são as seguintes:

- 1) Corpo de conhecimento atualizado sobre LPrS, obtido por meio de um mapeamento sistemático da literatura.
- 2) Corpo de conhecimento e análise de processos de teste de software, obtido por meio de um mapeamento sistemático da literatura.
- 3) Linha de processo de software para teste de aplicações móveis.
- 4) Ferramenta de apoio a linha de processo de software elaborada.

### **5.3. Limitações**

Algumas limitações foram observadas no decorrer da pesquisa:

- A ferramenta desenvolvida contém informações principais a respeito de teste de aplicações móveis, logo é necessária uma evolução da mesma em trabalhos futuros.
- Os participantes que avaliaram a proposta representam uma pequena porcentagem do total do público alvo da ferramenta. Assim é necessário ainda avaliar a utilização da ferramenta com mais especialistas.
- O método de validação utilizado da ferramenta proposta foi o *Survey*, por ser necessário um tempo maior para aplicação em um ambiente real na indústria, pois exige uma maior preparação e treinamento de envolvidos.
- A arquitetura da LPrS elaborada não foi avaliada por nenhum especialista na área, logo essa atividade deve ser realizada futuramente.

### **5.4. Trabalhos futuros**

- No decorrer da pesquisa foram publicados outros trabalhos relacionados a processo de teste de software. É necessário analisar o processo apresentado e estudar possíveis modificações no processo base.
- Durante a avaliação foi questionado a respeito da adoção de processos ágeis, sendo assim é necessária a inclusão de processos de teste ágeis a ferramenta, buscando uma maior aproximação ao ambiente real da indústria.
- Incluir outros processos na LPrS é um trabalho árduo e complexo, sendo necessário um estudo e adaptação dos processos, métodos e técnicas inseridos na ferramenta.
- A ferramenta não detém todas as informações a respeito de teste de aplicações móveis, logo é necessária uma expansão de seu conteúdo.
- Realizar um estudo de observação no ambiente industrial o que permitiria o amadurecimento da ferramenta.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ABRAN, A.; BOURQUE, P.; DUPUIS, R.; MOORE, J. W. **Guide to the Software Engineering Body of Knowledge**. Piscataway, NJ, USA: IEEE Press, 2004.
- ALEIXO, F. A.; FREIRE, M. A.; SANTOS, W. C.; KULESZA, U. A. **Automating the Variability Management, Customization and Deployment of Software Processes: A Model-Driven Approach**. International Conference Enterprise Information Systems (ICEIS). Beijing, China: Springer Heidelberg, 2011
- ARANGO, G.; PRIETO-DIAZ, R. **Introduction and Overview: Domain Analysis Concepts and Research Direction**. G.Arango, R.P.-D.A. (eds), Domain Analysis and Software Systems Modeling, IEEE Computer Society Press, p. 9–25, 1991.
- ARMBRUST, O.; KATAHIRA, M.; MIYAMOTO, Y.; MÜNCH, J.; NAKAO, H.; OCAMPO, A. **Scoping Software Process Models - Initial Concepts and Experience from Defining Space Standards**. International Conference Software Process: Improvement and Practice., v. 5007, p. 14, 2008.
- BARRETO, A.; DUARTE, E.; ROCHA, A. R.; MURTA, L. **Supporting the Definition of Software Processes at Consulting Organizations via Software Process Lines**. Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the, p. 15–24, 2010.
- BASIL, V. R.; ROMBACH, H. D. **Goal Question Metric Paradigm**. 2 Volume S ed. [s.l.] John Wiley & Sons, Inc, 1994.
- BAYER, J.; GIESE, C.; WEILAND, J. **Process Family Engineering**. n. 18, 2005.
- BENDRAOU, R.; JÉZÉQUEL, J.-M.; GERVAIS, M.-P.; BLANC, X. **A comparison of Six UML-Based Languages for Software Process Modeling**. IEEE Transactions on Software Engineering, v. 36, n. 5, p. 662–675, 2010.
- BENDRAOU, R.; SADOVYKH, A.; GERVAIS, M.-P.; BLANC, X. **Software Process Modeling and Execution: The UML4SPM to WS-BPEL Approach**. Conference on Software Engineering and Advanced Applications - EUROMICRO. Lubeck, Germany: IEEE Computer Society, 2007
- BERTOLLO, G.; SEGRINI, B.; FALBO, R. D. A. **Definição de Processos de Software em um**

- Ambiente de Desenvolvimento de Software Baseado em Ontologias**Simpósio Brasileiro de Qualidade de Software (SBQS). **Anais...**Vila Velha, Espírito Santo, Brasil.: 2006
- CARSTEN, B. **Carsten's Corner**. Revista Power Conversion and Intelligent Motion, p. 38, 1989.
- CARVALHO, D.; CHAGAS, L.; LIMA, A.; REIS, C. Software Process Lines: A Systematic Literature Review. In: Communications in Computer and Information Science. [s.l.] Springer International Publishing, 2014. p. 118–130.
- CHEN, G.; KOTZ, D. **A Survey of Context-Aware Mobile Computing Research**. Technical Report TR2000-381, v. 3755, n. TR2000-381, p. 1–16, 2000.
- CHOU, S, C. **A process modeling language consisting of high level UML diagrams and low level process language**. Journal of Object-Oriented Programming, v. 1, n. 4, p. 137–163, 2002.
- CMMI. **CMMI ® para Desenvolvimento – Versão 1.2**. 2006.
- CRESPO, A. N.; SILVA, O. J. DA; BORGES, C. A.; SALVIANO, C. F.; TEIVE, M. DE; JUNIOR, A.; JINO, M. **Uma Metodologia para Teste de Software no Contexto da Melhoria de Processo**. n. Simpósio Brasileiro de Qualidade de Software, p. 271–285, 2004.
- CUGOLA, G.; GHEZZI, C. **Software processes: A retrospective and a path to the future**. Journal of Software Process Improvement and Practice (SPIP), v. 4, p. 101–123, 1998.
- DAVIS, F. D.; BAGOZZI, R. P.; WARSHAW, P. R. **User Acceptance of Computer Technology: A Comparison of Two Theoretical Models**. Management Science, 1989.
- DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. **Introdução ao teste de software**. 2<sup>a</sup> ed. [s.l.] ELSEVIER, 2016.
- DI NITTO, E.; LAVAZZA, L.; SCHIAVONI, M.; TRACANELLA, E.; TROMBETTA, M. **Deriving executable process descriptions from UML**International Conference on Software Engineering - ICSE. **Anais...**Orlando, Florida: ACM, 2002
- DIAS-NETO, A. C.; TRAVASSOS, G. H. **Maraká : Infra-estrutura Computacional para Apoiar o Planejamento e Controle dos Testes de Software**Simpósio Brasileiro de Qualidade de Software (SBQS). **Anais...**Vila Velha, Espírito Santo, Brasil.: 2006
- DUSTIN, E. **Effective Software Testing: 50 Ways to Improve Your Software Testing**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

- FRANCH, X.; BOTELLA, P.; BURGUS, X.; RIBÓ, J. M. **ComProLab: A Component Programming Laboratory** Software Engineering and Knowledge Engineering - SEKE. *Anais...Madri*: 1997
- FUGGETA, A. **Software process: a roadmap** Proceedings of the Conference on The Future of Software Engineering. *Anais...Limerick, Ireland*: 2000
- GAO, J.; BAI, X.; TSAI, W.-T.; UEHARA, T. **Mobile Application Testing: A Tutorial**. *Computer*, v. 47, p. 46–55, 2014.
- GARCIA, C.; PALUDO, M.; MALUCELLI, A.; REINEHR, S. **A software process line for service-oriented applications**. Proceedings of the 30th Annual ACM Symposium on Applied Computing, p. 1680–1687, 2015.
- GRISS, M. L.; FAVARO, J.; D’ALESSANDRO, M. **Integrating feature modeling with the RSEB**. Proceedings. Fifth International Conference on Software Reuse (Cat. No.98TB100203), p. 76–85, 1998.
- IEEE, C. S. **Guide to the Software Engineering Body of Knowledge - SWEBOK**. Los Alamitos: CS Press, 2004.
- ISO/IEEE 29119-2, I. **INTERNATIONAL STANDARD ISO / IEC / IEEE 29119-2 Software and systems engineering — Software Testing**. v. 2013, 2013.
- ISO/IEEE 829-2008, I. **IEEE Std 829-2008, IEEE Standard for Software and System Test Documentation**. [s.l: s.n.]. v. 2008
- JAUFMAN, O.; MÜNCH, J. **Acquisition of a Project-Specific Process**. Proceedings of the 6th International Conference on Product-Focused Software Process Improvement, p. 328–342, 2005.
- KANG, K. C.; COHEN, S. G.; HESS, J. A.; NOVAK, W. E.; PETERSON, A. S. **Feature-Oriented Domain Analysis (FODA) Feasibility Study**. *Distribution*, v. 17, n. November, p. 161, 1990.
- KIRUBAKARAN, B.; KARTHIKEYANI, V. **Mobile application testing — Challenges and solution approach through automation**. 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, p. 79–84, 2013.
- KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. *Engineering*, v. 2, p. 1051, 2007.

- MAGDALENO, A. M. .; ARAUJO, R. M. .; WERNER, C. M. L. .; BATISTA, C. F. A. . **A practical experience of a software process line creation.** CIBSE 2015 - XVIII Ibero-American Conference on Software Engineering, p. 95–108, 2015.
- MAGDALENO, A. M.; ARAUJO, R. M.; WERNER, C. M. L. **COMPOOTIM: An approach to Software Process Composition and Optimization.** Congresso Ibero-Americano em Engenharia de Software (CIBSE),. Buenos Aires, Argentina: 2012
- MAGDALENO, A. M.; BARROS, M. DE O.; WERNER, C. M. L.; ARAUJO, R. M.; BATISTA, C. F. A. **Collaboration Optimization in Software Process Composition.** Journal of Systems and Software (JSS) - Special Issue Search Based Software Engineering (SBSE), 2014.
- MALHOTRA, N. **Pesquisa de marketing: uma orientação aplicada.** 4ª ed. Porto Alegre: Bookman, 2006.
- MARTÍNEZ-RUIZ, T.; GARCIA, F.; PIATTINI, M.; MÜNCH, J. **Modelling software process variability: an empirical study.** Institution of Engineering and Technology, IET Software., v. 5, n. 2, p. 172–187, 2011.
- MASI, E.; CANTONE, G.; MASTROFINI, M.; CALAVARO, G.; SUBIACO, P. **Mobile Apps Development: A Framework for Technology Decision Making.** (D. Uhler, K. Mehta, J. L. Wong, Eds.)MobiCASE. *Anais...*: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.Springer, 2012
- MENDES, I. N.; DIAS-NETO, A. C. **A Process-Based Approach to Test Usability of Multi-Platform Mobile Applications Usability Testing in Mobile Applications**Human-Computer Interaction International Conference. Toronto - Canada: 2016
- MILER, N. **A Engenharia de Aplicações no Contexto de Reutilização baseada em Modelos de Domínio.** [s.l.] COPPE/UFRJ, Rio de Janeiro, RJ, Brasil., 2000.
- MPS.BR. **MPS.BR -Melhoria de Processo do Software Brasileiro Guia Geral MPS de Software.** 2012.
- MUCCINI, H.; FRANCESCO, A.; ESPOSITO, P. **Software testing of mobile applications: Challenges and future research directions.** 7th International Workshop on Automation of Software Test (AST 2012), p. 29–35, 2012.
- MYERS, G. J. **The art of software testing.** New York, NY, USA: John Wiley & Sons, Inc, 1979.

- OLIVEIRA, R. F. **Formalização e Verificação de Consistência na Representação de Variabilidades**. [s.l.] COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2006.
- OMG, O. M. G. **Software & Systems Process Engineering Metamodel specification (SPEM) Version 2.0**.
- OSTERWEIL, L. **Software Processes are Software too** Proceedings of the 9th International Conference on Software Engineering. **Anais...**Monterey, USA: 1987
- PRESSMAN, R. S. **Software Engineering: A Practitioner's Approach**. 7th. ed. [s.l.] McGraw-Hill Education, 2009.
- ROMBACH, D. **Integrated software process and product lines**. International Software Process Workshop (SPW) 2005, Beijing, p. 83–90, 2005.
- SALEH, A. M. **Adoção de tecnologia: um estudo sobre a adoção de software livre nas empresas**. [s.l.] Universidade de São Paulo, 2004.
- SIMOS, M.; ANTHONY, J. **Weaving the model web: a multi-modeling approach to concepts and features in domain engineering**. Proceedings. Fifth International Conference on Software Reuse (Cat. No.98TB100203), p. 94–102, 1998.
- SOMMERVILLE, I. **Software Engineering**. 10th. ed. Pearson Education, 2015.
- SCAMPI. **Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition Document**. 2011.
- TEIXEIRA, E. N, WERNER, C. M. L. **OdysseyProcessReuse: Uma proposta de sistemática de engenharia de linha de processo de software baseada em um projeto arquitetural de componentes de processo de software**. Qualificação de Doutorado, COPPE, UFRJ, Rio de Janeiro, 2014.
- WASHIZAKI, H. **Building Software Process Line Architectures from Bottom Up**. Product Focused Software Process Improvement, v. 4034, p. 415–421, 2006.
- WASSERMAN, A. I. **Software engineering issues for mobile application development**. Proceedings of the FSE/SDP workshop on Future of software engineering research - FoSER '10, p. 397, 2010.

## APÊNDICE A – ARTIGOS SELECIONADOS NO MAPEAMENTO SISTEMÁTICO

- [1] ONSUK, O., WUTTIDITTACHOTTI, P., PRAKANCHAROEN, S., VALLIPAKORN, S. **A Sdlc Developed Software Testing Process Using Dmaic Model**. ARPN Journal of Engineering and Applied Sciences, v. 10, n. 3, p. 1008–1016, 2015.
- [2] ABDOU, T.; KAMTHAN, P. **A Knowledge Management Approach for Testing Open Source Software Systems**. 33rd IEEE -- International Performance Computing and Communications Conference, Texas, USA, n. 1, p. 4–5, 2014.
- [3] SULAIMAN, N. A.; KASSIM, M. **Developing A Customized Software Engineering Testing For Shared Banking Services ( Sbs ) System**. International Conference on System Engineering and Technology (ICSET), p. 132–137, 2011.
- [4] DESAI, A.; SHAH, S. **Knowledge management and software testing**. Proceedings of the International Conference & Workshop on Emerging Trends in Technology, n. ICWET, p. 767–770, 2011.
- [5] RODRIGUES, A.; PINHEIRO, P. R.; ALBUQUERQUE, A. **The Definiton of a Testing Process to Small-Sized Companies: The Brazilian Scenario**. 2010 Seventh International Conference on the Quality of Information and Communications Technology, n. July 2015, p. 298–303, 2010.
- [6] QIAN, Q., YINHUI, C., FENG, Q., XUESONG, Q. **Workflow-Based Process Management of Network Management Testing**. 2009 Third International Symposium on Intelligent Information Technology Application Workshops, p. 265–268, 2009.
- [7] KARINSALO, M.; ABRAHAMSSON, P. **Software Reuse and the Test Development Process: A Combined Approach**. Software Reuse: Methods, Techniques and Tools, p. 59–68, 2004.
- [8] KASOJU, A.; PETERSEN, K.; MÄNTYLÄ, M. V. **Analyzing an automotive testing process with evidence-based software engineering**. Information and Software Technology, v. 55, n. 7, p. 1237–1259, 2013.
- [9] KHAN, R. A.; CHOUDHARY, R. K. **Software Testing Process: A Prescriptive Framework**. SIGSOFT Softw. Eng. Notes, v. 36, n. 4, p. 1–5, 2011.
- [10] COLLINS, E. F.; DE LUCENA JR., V. F. **Software Test Automation Practices in Agile Development Environment: An Industry Experience**. Report Proceedings of the 7th

International Workshop on Automation of Software Test. Anais...: AST '12.Piscataway, NJ,  
USA: IEEE Press, 2012

## APÊNDICE B – ARTIGOS SELECIONADOS NA EXTENSÃO DO MAPEAMENTO SISTEMÁTICO

- [1] CASARE, S.; ZIADI, T.; BRANDAO, A.; GUESSOUM, Z. **Meduse: An Approach for Tailoring Software Development Process**. Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems, ICECCS, 2017.
- [2] KUHRMANN, M.; TERNITA, T.; FRIEDRICH, J.; RAUSCH, A. BROY, M. **Flexible software process lines in practice: A metamodel-based approach to effectively construct and manage families of software process models**. Journal of Systems and Software, 2016.
- [3] GONZALEZ, F.; SILVESTRE, L.; BASTARRICA, M.; SOLARI, M. **Template-Based vs. Automatic Process Tailoring**. Proceedings - International Conference of the Chilean Computer Science Society, SCCC, 2016.
- [4] TERCENIANI, M.; LANDRE, G.; PAIVA, D.; CAGNIN, M. **A plug-in for Eclipse towards supporting business process lines documentation**. Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA, 2016.
- [5] AYORA, C.; TORRES, V.; DE LA VARA, J.; PELECHANO, V. **Variability management in process families through change patterns**. Information and Software Technology, 2016.
- [6] KUHRMANN, M; MANDEZ FERNANDEZ, D.; TERNITA, T. **On the use of variability operations in the V-Modell XT software process line**. Journal of Software: Evolution and Process, 2016.
- [7] COGNINI, R.; CORRADINI, F.; POLINI, A.; RE, B. **Process Variability Modeling for Complex Organizations**. 3rd International Conference on Enterprise Systems ES, 2015.
- [8] LOIOLA, E.; DA SILVEIRA, D.; ARAUJO, J.; MOREIRA, A. **Business process families: A case study in the Brazilian public sector**. CEUR Workshop Proceedings, 2016.
- [9] SCHRAMM, J.; DOHRMANN, P.; KUHRMANN, M. **Development of flexible software process lines with variability operations: A longitudinal case study**. ACM International Conference Proceeding Series, 2015.
- [10] GARCIA, C.; MALUCELLI, A.; PALUDO, M.; REINEHR, S. **A software process line for service-oriented applications**. Proceedings of the ACM Symposium on Applied Computing, 2015.
- [11] TERCENIANI, M.; PAIVA, D.; LANDRE, G.; CAGNIN, M. **BPMN\* - A Notation for Representation of Variability in Business Process Towards Supporting Business Process Line Modeling**. Proceedings of the International Conference on Software Engineering and

Knowledge Engineering SEKE, 2015.

[12] TEIXEIRA, E.; DE MELLO, R.; MOTTA, R.; WERNER, C.; VASCONCELOS, A. **Verification of software process line models: A checklist-based inspection approach.** CIBSE 2015 - XVIII Ibero-American Conference on Software Engineering, 2015.

[13] ZHANG, H.; WANG, F.; ZHANG, Y.; ZHOU, Z. **Requirement-driven adaptive business processes.** 14th International Symposium on Communications and Information Technologies ISCIT, 2015.

[14] MAGDALENO, A.; ARAUJO, R.; WERNER, C.; BATISTA, C. **A practical experience of a software process line creation.** CIBSE 2015 - XVIII Ibero-American Conference on Software Engineering, 2015.

[15] BRONDANI, C.; BERTUOL, G.; FONTOURA, L. **Quality evaluation of artifacts in tailored software process lines.** Proceedings of the International Conference on Software Engineering and Knowledge Engineering SEKE, 2015.

[16] GALLINA, B.; KASHIYARANDI, S.; MARTIN, H.; BRAMBERGER, R. **Modeling a safety- and automotive-oriented process line to enable reuse and flexible process derivation.** Proceedings - IEEE 38th Annual International Computers Software and Applications Conference Workshops COMPSACW, 2014.

[17] KUHRMANN, M.; FERNANDEZ, D.; TERNITA, T. **Realizing software process lines: Insights and experiences.** ACM International Conference Proceeding Series, 2014.

[18] DIAS DE CARVALHO, D.; CHAGAS, L.; REIS, C. **Definition of software process lines for integration of Scrum and CMMI.** Proceedings of the 2014 Latin American Computing Conference CLEI, 2014.

[19] BASTARRICA, M.; SIMMONDS, J.; SILVESTRE, L. **Using megamodeling to improve industrial adoption of complex MDE solutions.** 6th International Workshop on Modeling in Software Engineering MiSE, 2014.

[20] SCHUNSELAAR, D.; VERBEEK, H.; VAN DER AALST, W.; REIJERS, H. **Petra: A tool for analysing a process family.** CEUR Workshop, 2014.

[21] FENG, Z.; CHIU, D.; PENG, R.; GONG, P.; HE, K.; HUANG, Y. **Facilitating Cloud Process Family Co-evolution by Reusable Process Plug-in: An Open-source Prototype.** IEEE Transactions on Services Computing, 2015.

[22] SIMMONDS, J.; PEROVICH, D.; BASTARRICA, M. C.; SILVESTRE, L. **A megamodel for Software Process Line modeling and evolution.** 18th International Conference on Model Driven Engineering Languages and Systems (MODELS), 2015.

[23] GALLINA, B.; FABRE, L. **Benefits of security-informed safety-oriented process line engineering**. 34th Digital Avionics Systems Conference (DASC), 2015.

## APÊNDICE C – MODELO DE CARACTERÍSTICAS

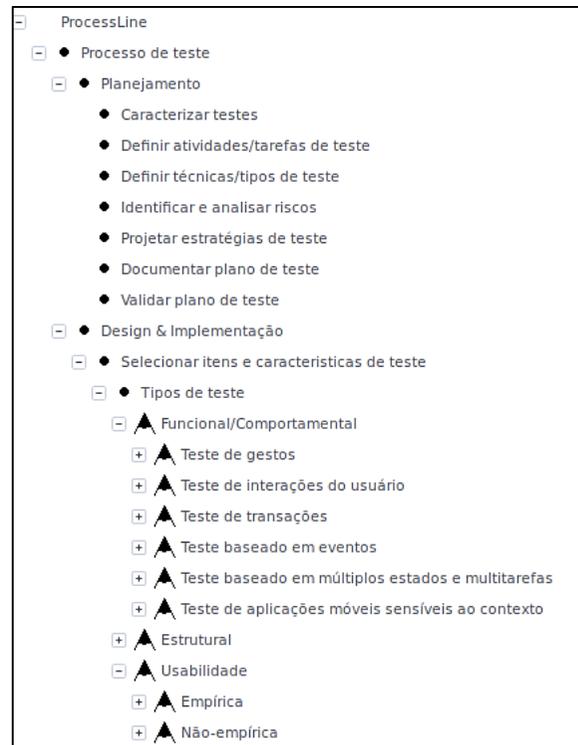


Figura 47. Modelo de características da MTPL



Figura 48. Restrições da MTPL

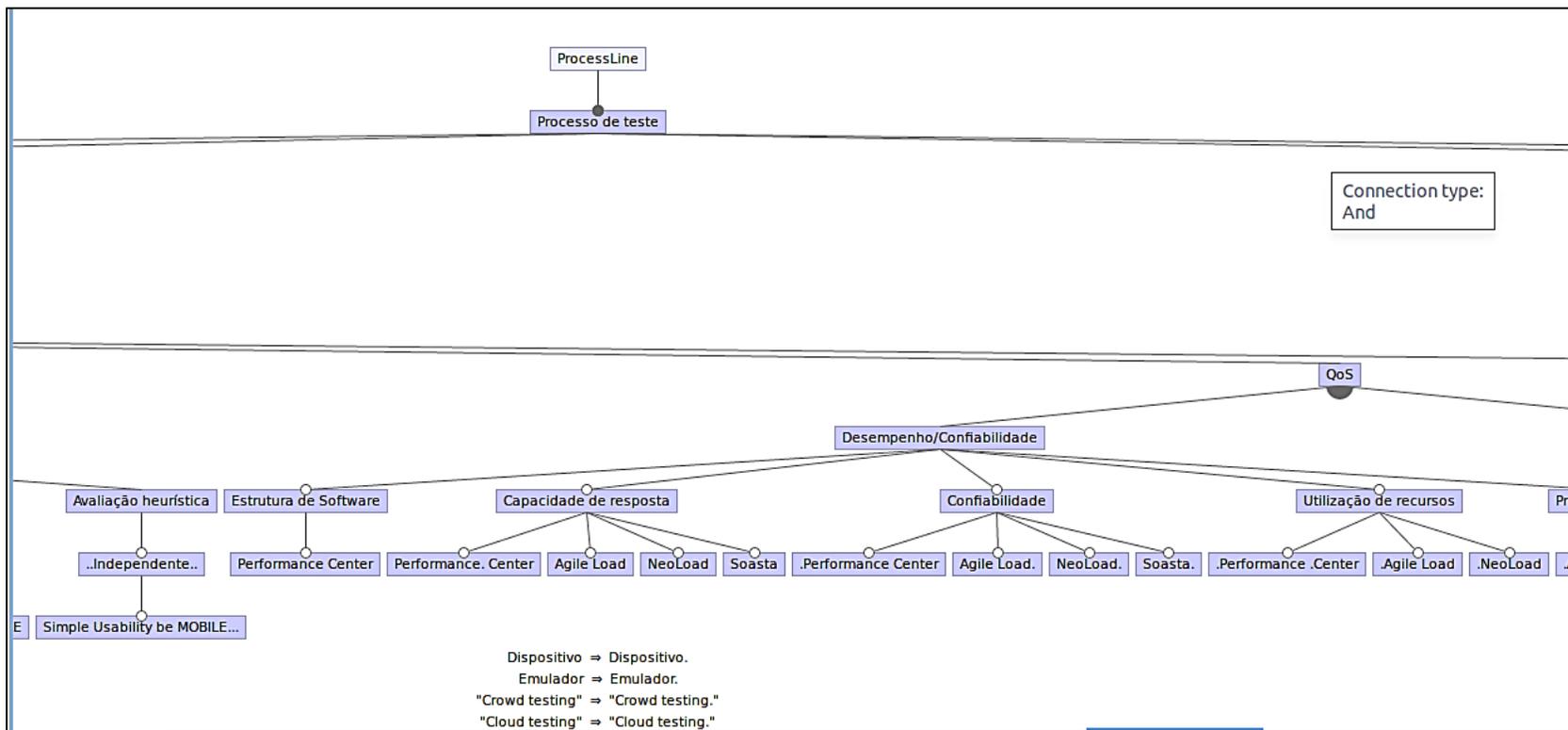


Figura 49. Diagrama de características da MTPL

## APÊNDICE D – FERRAMENTAS DA MTPL

<b>Ferramenta</b>	<b>Ambiente de execução</b>	<b>Plataforma de <u>Execução</u></b>	<b>Abordagem de teste</b>
MITE	Emulador e dispositivo	Android, iOS e Windows Phone	Funcional / Comportamental
Sikuli	Emulador e dispositivo	Android, iOS e Windows Phone	Funcional / Comportamental
Seledroid	Emulador e dispositivo	Android, iOS e Windows Phone	Funcional / Comportamental
Coded UI	Emulador e dispositivo	Windows Phone	Funcional / Comportamental
EggPlant	Emulador	Android e iOS	Funcional / Comportamental
MonkeyRunner	Emulador e dispositivo	Android	Funcional / Comportamental
Robotium	Emulador e dispositivo	Android	Funcional / Comportamental
Selenium Android Webdrive	Emulador e dispositivo	Android	Funcional / Comportamental
AM-TaaS	Nuvem	Android	Funcional / Comportamental
Sauce Labs	Nuvem	Android, iOS e Windows Phone	Funcional / Comportamental
Perfecto Mobile	Nuvem	Android, iOS e Windows Phone	Funcional / Comportamental

Testdroid	Nuvem	Android e iOS	Funcional / Comportamental
AWS Device Farm	Nuvem	iOS e Android	Funcional / Comportamental
Xamarin Test Cloud	Nuvem	iOS e Android	Funcional / Comportamental
Firebase Test Lab	Nuvem	Android	Funcional / Comportamental
Monkey	Emulador e dispositivo	Android	Funcional / Comportamental
Sapienz	Emulador	Android	Funcional / Comportamental
Espresso	Emulador e dispositivo	Android	Funcional / Comportamental
Barista	Emulador, dispositivo e larga escala	Android	Funcional / Comportamental
UserLytics	Emulador e dispositivo	Android e iOS	Usabilidade
APPLAUSE	Emulador	Android e iOS	Usabilidade
APPSEE	Emulador	Android e iOS	Usabilidade
Arise	Emulador	Android e iOS	Usabilidade
Keynote	Emulador e dispositivo	Android e iOS	Usabilidade
UserZoom	Dispositivo	Android e iOS	Usabilidade

Mr. Tappy	Dispositivo	Android e iOS	Usabilidade
Apptimize	Emulador e dispositivo	Android e iOS	Usabilidade
SimpleUsability's beMobile	Emulador e dispositivo	Android, iOS e Windows Phone	Usabilidade
Apperian	Emulador	Android e iOS	Usabilidade
AgileLoad	Emulador	-	Qualidade de serviço
NeoLoad	Emulador e dispositivo	-	Qualidade de serviço
SOASTA	Emulador	-	Qualidade de serviço
App Inspector	-	-	Qualidade de serviço
Fibedroid	-	-	Qualidade de serviço

## APÊNDICE E - FORMULÁRIO DE CARACTERIZAÇÃO DO ESPECIALISTA

**\* Qual papel mais se aproxima de suas atribuições?**

Gerente de teste / Gerente de qualidade  
 Analista de testes / Líder técnico de testes  
 Outro (especifique)

**Figura 50.** Formulário de caracterização - Papel

**\* Qual seu nível de conhecimento relacionado a cada um dos assuntos abaixo?**

	Desconheço o assunto	Superficial	Bom	Excelente
Elaboração de Linha de Processo de Software (LPrS)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Utilização de Linha de Processo de Software (LPrS)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Processo de teste de software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ISO/IEEE 29119	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Teste de aplicações móveis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Figura 51.** Formulário de caracterização - nível de conhecimento

**\* Quanto tempo de experiência em teste de software?**

Mais de 10 anos  
 5 - 10 anos  
 2 - 5 anos  
 Menor que 2 anos

**Figura 52.** Formulário de caracterização - tempo de experiência

## APÊNDICE F – FORMULÁRIO DE AVALIAÇÃO DA FERRAMENTA MTPLT

* Realizar a seguinte atividade foi fácil					
	Discordo plenamente	Discordo parcialmente	Neutro	Concordo parcialmente	Concordo plenamente
Derivar um processo de teste funcional utilizando abordagem de teste de gestos de forma não automatizado para qualquer plataforma e ambiente de execução	<input type="radio"/>				

**Figura 53.** Realização da tarefa específica do grupo.

	Discordo plenamente	Discordo parcialmente	Neutro	Concordo parcialmente	Concordo plenamente
Aprender a utilizar o MTPL foi fácil.	<input type="radio"/>				
Utilizar o MTPL é importante e adicionar valor ao meu trabalho.	<input type="radio"/>				
Eu gostaria de trabalhar com o MTPL.	<input type="radio"/>				
O acesso ao MTPL é simples.	<input type="radio"/>				
O MTPL é útil na derivação de processos de teste para aplicações móveis.	<input type="radio"/>				
Usar o MTPL produz os resultados que espero de uma sistematização de processos.	<input type="radio"/>				
Usar o MTPL minimizou o tempo de definição de processo de testes e ferramentas a serem utilizadas.	<input type="radio"/>				
Minha interação com o MTPL é clara e compreensível.	<input type="radio"/>				
Compreendo claramente as informações apresentadas no relatório provido pelo MTPL.	<input type="radio"/>				

**Figura 54.** Perguntas de avaliação da ferramenta

Preencha o campo abaixo caso tenha alguma sugestão ou crítica a respeito da ferramenta MTPL ( <i>Mobile Testing Process Line Tool</i> ).
<div style="border: 1px solid black; width: 100%; height: 100%;"></div>

**Figura 55.** Campo de críticas e sugestões da ferramenta