UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

PRÓ-REITORIA DE ENSINO DE GRADUAÇÃO

DEPARTAMENTO DE COMPUTAÇÃO (DC)

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RODRIGO CUNHA ALVES MOREIRA

Uma Proposta de Taxonomia de Casos de Teste de Software de Caixa Preta

TRABALHO DE CONCLUSÃO DE CURSO

Recife

2019

RODRIGO CUNHA ALVES MOREIRA

Uma Proposta de Taxonomia de Casos de Teste de Software de Caixa Preta

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação, como parte dos requisitos necessários à obtenção do título de Bacharel em Ciência da Computação.

Orientadora: Ana Paula Carvalho Cavalcanti Furtado

Recife
2019

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

M838p Moreira, Rodrigo Cunha Alves.
Uma proposta de taxonomia de casos de teste de software de caixa preta / Rodrigo Cunha Alves Moreira. – Recife, 2019.
61 f.: il.

Orientador(a): Ana Paula Carvalho Cavalcanti Furtado.
Trabalho de Conclusão de Curso (Graduação) – Universidade Federal Rural de Pernambuco, Departamento de computação, Recife, BR-PE, 2019.
Inclui referências e apêndice(s).

1. Teste de software 2. Casos de teste 3. Taxonomia I. Furtado, Ana Paula Carvalho Cavalcanti, orient. II. Título

CDD 004

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

http://www.bcc.ufrpe.br

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por Rodrigo Cunha Alves Moreira às 09 horas do dia 15 de janeiro de 2019, no Auditório do DEINFO, como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado " **Uma Proposta de Taxonomia de Casos de Teste de Software de Caixa Preta** ", orientado por Ana Paula Carvalho Cavalcanti Furtado e aprovado pela seguinte banca examinadora:

Ana Paula Carvalho Cavalcanti Furtado

Sidney de Carvalho Nogueira
DC/UFRPE

À memória de Paulo Ricardo Pimentel Cunha e Maria José Freire Moreira

Agradecimentos

Agradeço primeiramente a Deus por ter me dado saúde e força na vida para que eu pudesse concluir este trabalho.

Agradeço a minha orientadora, professora Ana Paula, por ter me guiado em todas as etapas do desenvolvimento deste trabalho.

Agradeço aos meus pais Ricardo e Roberta, que sempre fizeram de tudo para que eu pudesse alcançar os meus objetivos pessoais e profissionais.

Agradeço ao meu irmão Rafael, que sempre esteve ao meu lado e me serviu de exemplo para que eu concluísse a graduação.

Agradeço a minha namorada Marina, por sempre ter me apoiado e me aconselhado na conclusão deste trabalho.

Agradeço aos meus amigos da universidade Leonardo, Daniel, Pedro, Thomas, Dennys, Thiago e Ítalo, que puderam compartilhar momentos felizes e difíceis durante a graduação.

Agradeço aos participantes do grupo focal, que se dispuseram e gastaram um pouco do seu tempo com o intuito de contribuir com a pesquisa e acrescentando assim, uma melhor qualidade ao trabalho desenvolvido.

Por fim, agradeço a todos que direta ou indiretamente contribuíram para que fosse possível a realização deste trabalho, o meu muito obrigado.

“Os computadores são incrivelmente rápidos, precisos e burros; os homens são incrivelmente lentos, imprecisos e brilhantes; juntos, seus poderes ultrapassam os limites da imaginação.”

(Albert Einstein)

Resumo

Nos últimos anos houve um grande crescimento de empresas que trabalham com desenvolvimento de software, da mesma maneira, a exigência pela qualidade nestes softwares produzidos também aumentou. Consequentemente, o teste de software tem um papel muito importante na garantia da qualidade de um software. Casos de teste é um conceito fundamental para teste de software; consiste em um conjunto de entradas, condições e resultados esperados com o objetivo de validar uma função específica ou requisito do software. No desenvolvimento do software são criados inúmeros casos de teste, e isto pode deixar os engenheiros de testes confusos. Com isso, este trabalho tem como objetivo desenvolver uma classificação de casos de testes de software de caixa preta com o intuito de auxiliar os engenheiros de testes em suas execuções de planos de teste. Para realizar isto, foi feito uma revisão bibliográfica exploratório com o intuito de buscar uma fundamentação teórica, embasamento da pesquisa e análise dos trabalhos relacionados. Foi desenvolvido então uma taxonomia, que é uma coleção de termos, de um vocabulário controlado, organizados em uma estrutura hierárquica. Após o desenvolvimento da taxonomia, foi realizado uma validação dessa proposta por meio da técnica de grupo focal. A partir desta validação, foi possível obter feedbacks positivos com relação a proposta, constatar a relevância da pesquisa e mostrar indícios de sua validade.

Palavras-chave: teste de software, casos de teste, taxonomia.

Abstract

In recent years, there has been a huge growth of companies that work with software development, in the same way, the demand for quality in these softwares has also increased. Because of this, software testing plays a very important role in ensuring the quality of software. When it comes to software testing, it is important to mention test cases, which are a set of inputs, conditions, and expected results for the purpose of validating a specific function or software requirement. Countless test cases are created during the development of a software, and it may confuse the test engineers. Thus, the objective of this work was to develop a classification of black box software test cases with the purpose of assisting the test engineers in their test plan executions. To accomplish this, an exploratory literature review was done with the intention of seeking a theoretical basis and analysis of related works. A taxonomy has been developed, which is a collection of terms, from a controlled vocabulary, organized in a hierarchical structure. After the development of this taxonomy, a validation of this proposal was accomplished through the focal group technique. From this validation, it was possible to obtain positive feedback regarding the proposal, verify the relevance of the research and to show evidence of its validity.

Keywords: software testing, test cases, taxonomy.

Lista de ilustrações

Lista de quadros

Lista de abreviaturas e siglas

CMMI	Modelo Integrado de Maturidade em Capacitação (do inglês, Capability Maturity Model Integration)
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
MBST	Teste de Segurança Baseado em Modelo (do inglês, Model-based Security Testing)
MBT	Teste Baseado em Modelo (do inglês, Model-based Testing)
MPS.BR	Melhoria do Processo de Software Brasileiro
QA	Garantia de Qualidade (do inglês, Quality Assurance)
TI	Tecnologia da Informação
UML	Linguagem de Modelagem Unificada (do inglês, Unified Modeling Language)

Sumário

1 INTRODUÇÃO

Este capitulo descreve o contexto deste trabalho, assim como a motivação que levou a realizá-lo, problema da pesquisa, objetivos e a estrutura deste documento.

1.1 Contexto

É difícil imaginar uma empresa, independente do setor, que não use software em seus processos. Atualmente existe um número grande de empresas (pequenas e grandes) que produzem software se comparado há uma ou duas décadas. De acordo com um estudo realizado pela ASSOCIACÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE (2017), o mercado brasileiro de TI, incluindo hardware, software, serviços e exportações de TI, movimentou 39,6 bilhões de dólares em 2016, o que representa 2,1% do PIB brasileiro e 1,9% do total de investimentos de TI no mundo. Se considerarmos a nível mundial, esse valor é de 2,03 trilhões de dólares. O estudo ainda indica que de 2006 a 2016, os principais indicadores do mercado brasileiro de software e serviços teve um aumento de quase 300%, passou de cerca de 7 bilhões para quase 20 bilhões. Neste mesmo estudo, foram identificados no ano de 2016, cerca de, 15.700 empresas atuando no setor de Software e Serviços no Brasil.

Entretanto, um software não pode ser desenvolvido de qualquer maneira, ele deve atingir um certo nível de qualidade para atender a satisfação dos seus usuários. A elaboração de um software deve conter não só atividades que contribuem diretamente para a sua construção, mas também atividades que visam checar a qualidade do processo de desenvolvimento e dos artefatos produzidos. De acordo com Baresi e Pezzè (2006), o desenvolvimento de grandes sistemas de software é complexo e propício a erros durante o seu processo. Erros podem ocorrer em qualquer fase do desenvolvimento, e eles devem ser identificados e removidos o mais rápido possível. Engenheiros de qualidade devem se envolver no processo de desenvolvimento para identificar os requisitos de qualidade e estimar o seu impacto durante o processo de desenvolvimento. O processo de qualidade faz parte de todo o ciclo de vida de um software e não apenas em uma fase. Inicia desde o estudo de viabilidade até a manutenção do software após a entrega. Problemas encontrados na fase inicial terão um custo bem menor se comparado com problemas encontrados em fases posteriores. Quanto mais demorado for para encontrar um defeito, mais custoso será para o projeto (BERTOLINO, 2007).

A Classificação é o processo pelo qual ideias e objetos são reconhecidos, diferenciados e classificados. Um tipo de classificação é a taxonomia. A definição exata

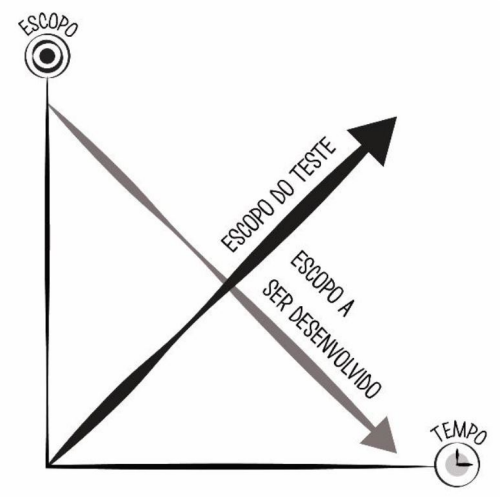
de taxonomia varia de fonte para fonte, mas o núcleo permanece o mesmo: O termo vem da área da biologia e é definida como a concepção, nomeação e classificação de grupos ou organismos (WILKINS, 2011). Na área de software é comum usar a taxonomia como ferramenta de classificação, pois ela permite organizar, dentro de um domínio ou área de conhecimento, diferentes termos integrados, fornecendo uma estrutura definida que fornece um mecanismo para identificar, atribuir e executar ações em um domínio, de forma clara, fornecendo soluções e gerando conhecimento (VILLALÓN et al., 2015). Na área de teste, a classificação é também de grande importância, pois irá permitir entender melhor os diferentes tipos de testes existentes, e com isso melhorar as tomadas de decisões dos engenheiros de testes com relação a quais tipos de testes utilizar em seus projetos.

Nesta pesquisa será proposto uma taxonomia com o intuito de auxiliar os engenheiros de testes em suas execuções. No momento da criação dos casos de teste, os engenheiros de testes poderão utilizar os atributos da taxonomia para rotular seus casos de testes. Desta forma, ficará mais fácil para encontrarem casos de teste que pertencem a um determinado domínio.

1.2 Motivação

Como mencionado na seção anterior, o número de empresas de software tem crescido muito. Para que esse crescimento continue, é necessário cada vez mais investimentos na qualidade dos softwares produzidos, o que implica dizer também que é necessário investimento em teste de software. De acordo com Baresi e Pezzè (2006), casos de teste podem e devem ser gerados assim que as especificações do software estão disponíveis, desta forma, softwares complexos terão cada vez mais testes enquanto o número de funcionalidades for aumentando. O escopo a ser desenvolvido em um projeto é inversamente proporcional ao escopo do teste, uma vez que as funcionalidades entregues vão se integrando ao que já tem sido produzido, fazendo com que uma maior quantidade de requisitos a serem testados sejam gerados (FURTADO, 2017) . O gráfico da Figura 1 representa isso.

Figura 1 – Escopo do teste e do software



FURTADO (2017)

Em certo ponto, haverá tantos casos de teste que deixará os engenheiros de testes confusos em saber quais testes usarem para as suas próximas execuções. Com isso, torna-se necessário uma classificação de casos de teste, com o intuito de diminuir esse esforço. Villalón et al. (2015) citam benefícios de uma taxonomia: ajudar a caracterizar os recursos disponíveis, reduzir o tempo e custo das execuções, aumentar a eficácia dos testes, permitir uma identificação clara e permitirá o gerenciamento do conhecimento do grupo de testes. Uma classificação pode permitir que se tenha um melhor entendimento sobre determinada área de teste, e com isso, servir como uma diretriz para decidir qual abordagem de teste se adequa a uma circunstância específica (FELDERER; AGREITER; ZECH, 2011).

1.3 Problemática

De acordo com Bertolino (2007), o processo de teste pode ser bastante custoso. Quanto mais requisitos e funcionalidades um projeto tem, maior será o número de testes que precisam ser executados para poder validar esse software com o intuito de garantir que o mesmo cumpra com todos os seus requisitos e funcionalidades. É normal, em sua fase de desenvolvimento, um software passar por diversas modificações, e essas modificações também precisam ser testadas (LAWANNA, 2015). Ainda de acordo com Bertolino (2007), executar todos os testes requer muito tempo e custo, portanto, faz se necessário selecionar uma porção menor de casos de teste.

De acordo com Engström, Runeson e Skoglund (2010), teste é muito importante, até mesmo crucial, para organizações que tem uma grande parcela de seus custos em desenvolvimento de software. Isto inclui, entre outras tarefas, determinar quais casos de teste devem ser reexecutados de forma que possa validar o comportamento do

software produzido.

De acordo com Lawanna (2015), um dos problemas da área de teste de software é selecionar os casos de teste mais adequados com relação ao tamanho do software. Se o tamanho for muito grande, isto pode afetar todo o desempenho do ciclo de vida do desenvolvimento do software.

Com isso, temos o seguinte problema: **Como classificar os casos de teste com o intuito de servir como um direcionamento aos engenheiros de testes em suas atividades?**

1.4 Objetivos

Esta seção apresenta os objetivos, geral e específicos, deste trabalho.

1.4.1 Objetivo Geral

Desenvolver uma classificação de casos de teste de software com o intuito de auxiliar os engenheiros de testes em suas execuções.

1.4.2 Objetivos Específicos

1) Levantar um conjunto de atributos, que compõem os casos de teste, para a classificação.

2) Coletar feedback da classificação de casos de teste de software proposta.

1.5 Estrutura do Trabalho

Este documento está dividido em 6 capítulos principais.

O primeiro capítulo contém a introdução da pesquisa, onde é feito uma contextualização do trabalho, além da justificativa, motivação e os seus objetivos.

O segundo capítulo é composto pela metodologia do trabalho. É mostrado todo o passo a passo realizado para desenvolver este trabalho, tais como suas fases e abordagens utilizadas.

O terceiro capítulo é formado pela revisão bibliográfica. Neste capítulo é mostrado todo o referencial teórico que serviu como base para esta pesquisa e que de certa maneira contribuem para o entendimento sobre os tópicos abordados por este trabalho.

O quarto capítulo contém os trabalhos relacionados. Neste capítulo é feito uma análise, com uma breve descrição, de alguns trabalhos que tem relação com o tema desta pesquisa.

O quinto capítulo é composto pela proposta do trabalho. Neste capítulo é mostrado a taxonomia de casos de teste de software de caixa preta, contendo a visão geral da proposta e passando por toda a sua decomposição.

O sexto capítulo é o da técnica do grupo focal. Neste capítulo é apresentado o planejamento e execução desta técnica de pesquisa, assim como a análise de dados obtidos e as suas limitações e ameaças à validade

Por fim, o sétimo capítulo contém a conclusão do trabalho. É mostrado as considerações finais e contribuições deste trabalho, assim como os trabalhos futuros.

2 METODOLOGIA

Este capítulo apresenta a metodologia adotada neste trabalho. Este estudo foi realizado através de uma abordagem exploratória que de acordo com Stebbins (2001), é quando a pesquisa tem como objetivo familiarizar-se com um fenômeno ou adquirir um novo conhecimento a fim de formular um problema ou desenvolver hipóteses.

2.1 Método da Pesquisa

No que diz respeito à abordagem de investigação, esta pesquisa seguiu um comportamento qualitativo. A pesquisa qualitativa é um método científico de observação para coletar dados não numéricos (BABBIE, 2014). Este tipo de pesquisa “refere-se aos significados, definições de conceitos, características, metáforas, símbolos e descrição das coisas” e não às suas “contagens ou medidas” (BERG; LUNE, 2012).

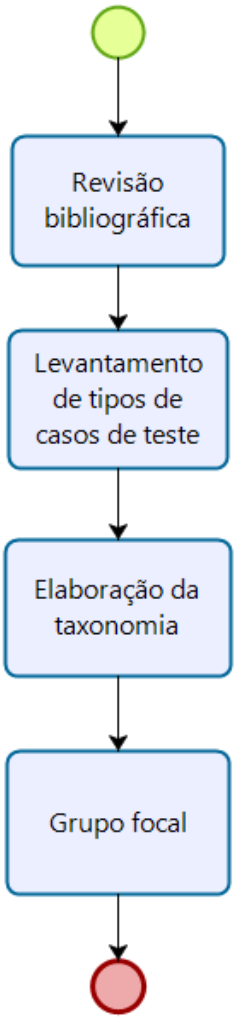
Abordagens de pesquisa qualitativa são empregadas em muitas disciplinas acadêmicas, focando particularmente em elementos humanos das ciências sociais e naturais (GIVEN, 2008). Em contextos menos acadêmicos, as áreas de aplicação incluem pesquisa qualitativa de mercado, negócios, demonstrações de serviço por organizações sem fins lucrativos e jornalismo (DENZIN; LINCOLN, 2005).

Como campo de estudo, as abordagens qualitativas incluem conceitos e métodos de pesquisa de vários campos acadêmicos estabelecidos. O objetivo de um projeto de pesquisa qualitativa pode variar de acordo com a formação disciplinar, como um psicólogo que busca uma compreensão profunda do comportamento humano e as razões que governam tal comportamento, por exemplo. Os métodos qualitativos são os melhores para pesquisar muitas das questões do “por que” e “como” da experiência humana (GIVEN, 2008).

2.2 Fases da Pesquisa

A pesquisa foi dividida em 4 principais etapas conforme mostrado na Figura 2.

Figura 2 – Etapas da pesquisa



Elaborado pelo autor

2.2.1 Revisão bibliográfica

De acordo com Freitas (2016), a revisão bibliográfica tem um papel fundamental no encaminhamento adequado de um problema de pesquisa, ela é de grande importância para poder estruturar a base de uma pesquisa. Ainda de acordo com Freitas (2016), é a partir da revisão bibliográfica que o pesquisador vai poder definir com mais precisão o objetivo do seu estudo, selecionando a literatura relevante para a sua pesquisa.

A primeira fase da pesquisa foi destinada a reunir materiais (revistas científicas, livros, trabalhos e artigos acadêmicos) sobre o tema relacionado, com o intuito de criar um referencial teórico dos principais tópicos que abordam este trabalho, assim como buscar por trabalhos relacionados. A revisão bibliográfica foi realizada de maneira

ad-hoc e foi de grande importância (e serviu como base) para o desenvolvimento deste trabalho.

2.2.2 Levantamento de tipos de casos de teste

Com base nos diversos materiais coletados na fase da revisão bibliográfica, foi realizado um levantamento de tipos de casos de teste de software de caixa preta.

2.2.3 Elaboração da taxonomia

A partir do levantamento realizado na etapa anterior, foi desenvolvido a proposta deste trabalho, uma taxonomia de casos de teste de software de caixa preta.

2.2.4 Grupo focal

De acordo com Caplan (1990), os grupos focais são pequenos grupos de pessoas reunidas para avaliar conceitos ou identificar problemas. A entrevista de grupo focal é uma técnica qualitativa que pode ser usada sozinha ou com outras técnicas qualitativas ou quantitativas (VAUGHN; SCHUMM; SINAGUB, 1996).

Segundo Kontio (2004), os grupos focais têm como objetivo obter as percepções dos membros do grupo em uma área de interesse definida. Ainda de acordo com Kontio (2004), os membros são selecionados com base em suas características individuais relacionadas ao tópico da sessão.

No contexto deste trabalho, o objetivo do grupo focal foi validar a proposta apresentada com o intuito de coletar sugestões de melhorias. Com isso, foi realizado as seguintes etapas como base em Kontio (2004):

1) Definir o problema de pesquisa
2) Planejar o grupo focal
3) Selecionar os participantes
4) Conduzir a sessão do grupo focal
5) Analisar e reportar dados coletados

Todas essas etapas estão detalhadas no Capítulo 6.

2.3 Considerações finais

Neste capítulo foi demonstrado toda a metodologia realizada para o desenvolvimento deste trabalho. Foi apresentado desde a fase inicial, onde houve coleta de informações e trabalhos relevantes, até a parte da validação da proposta apresentada, por meio do grupo focal.

3 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta o referencial teórico deste trabalho, fornece uma visão geral sobre qualidade de software com foco em teste de software e sobre classificação.

3.1 Qualidade de software

Qualidade é definida de várias maneiras por diferentes autores, e pode ter significados diferentes dependendo da área de conhecimento. A ideia de qualidade é aparentemente intuitiva, entretanto, quando analisado mais longamente, o conceito se revela complexo. Definir qualidade para estabelecer objetivos é, assim, uma tarefa menos trivial do que aparenta (KOSCIANSKI; SOARES, 2007).

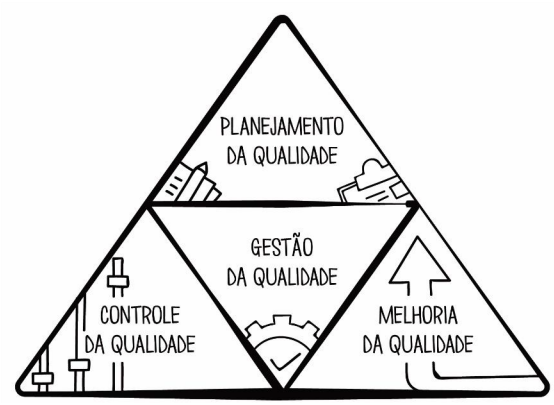
Organizações internacionais tem diferentes maneiras para definir qualidade. De acordo com GERMAN INDUSTRY STANDARD DIN 55350 (2004), qualidade compreende todas as características e funcionalidades significantes de um produto ou uma atividade relacionada com a satisfação de determinados requisitos. ANSI STANDARD (ANSI/ASQC) (1994) define qualidade como sendo a totalidade de funcionalidades e características de um produto ou serviço que atende à sua capacidade de satisfazer determinadas necessidades. Segundo IEEE STANDARD (2014), qualidade é o grau em que um produto ou processo atende aos requisitos estabelecidos, entretanto, qualidade depende do grau em que esses requisitos estabelecidos representam com precisão as necessidades, desejos e expectativas das partes interessadas (stakeholders).

W. Edwards Deming é talvez o especialista em qualidade mais conhecido do mundo. Ele foi fundamental na revitalização industrial do Japão no pós-guerra. Posteriormente, suas ideias foram cada vez mais adotadas na indústria nos Estados Unidos e em outros países. A partir de 1950, ele fez uma série de palestras para a alta gerência no Japão sobre o controle estatístico de processos (CHANDRUPATLA, 2009). Deming criou um método para a administração da qualidade, no qual, expôs catorze princípios, que são: Constância de propósito, adotar uma nova filosofia, cessar a dependência da inspeção em massa, extinguir a aprovação de orçamentos com base nos preços, melhoria contínua no sistema produtivo, instituir o treinamento, adotar e instituir a liderança, afastar o medo, derrubar barreiras entre os setores, eliminar “slogans” e metas, eliminar quotas numéricas, orgulho da mão de obra, estimular a formação e o aprendizado, e tomar iniciativa para realizar a transformação. A indústria japonesa adotou seus métodos que resultaram em uma melhoria significativa na qualidade (DEMING, 1982).

Juran (1998) deu duas definições sobre qualidade, e a que se encaixa no contexto desta pesquisa, é a de que qualidade significa liberdade de erros que exigem

retrabalho ou que resultam em insatisfação do cliente, reclamações dos clientes, e assim por diante. Desta forma, o significado de qualidade é orientado a custos, e quanto maior a qualidade, o custo é menor. Joseph Juran também defendia que a gestão qualidade se divide em três pontos fundamentais: (1) planejamento, (2) melhoria e (3) o controle de qualidade, mais conhecidas por “trilogia de Joseph Juran” e representada na figura 3.

Figura 3 – Trilogia de Juran



(JURAN, 1989)

Crosby (1979) definiu qualidade como sendo a conformidade com os requisitos. A definição por ele pressupõe que as especificações e requisitos já foram desenvolvidos, a próxima coisa a procurar é a conformidade com esses requisitos. Além disso, ele também incluiu outros três princípios: (1) o sistema de gestão é a prevenção, (2) o padrão de desempenho é o zero defeitos e (3) o sistema de medição é o custo da não conformidade.

Armand V. Feigenbaum foi uma figura central na história da qualidade. Ele se destacou pela publicação do seu livro em 1951 intitulado Controle de Qualidade: Princípios, Prática e Administração (do inglês, Quality Control: Principles, Practice and Administration), e que em 1961 foi publicado novamente pelo nome de Controle Total de Qualidade(do inglês, *Total Quality Control)*. Ele propôs um processo de três etapas para melhoria de qualidade: (1) liderança de qualidade, (2) tecnologia de qualidade e (3) comprometimento organizacional. O controle de qualidade total é um sistema eficaz para integrar os esforços de desenvolvimento de qualidade, manutenção de qualidade e melhoria de qualidade dos vários grupos em uma organização, permitindo que a produção e o serviço operem no nível mais econômico para alcançar a satisfação total do cliente (FEIGENBAUM, 1991).

3.2 Teste de Software

O teste de software é uma das atividades da engenharia de software que está diretamente ligada à qualidade. Ele tem um papel muito importante, pois poderá contribuir para melhorar a qualidade, diminuir os custos, evitar retrabalho, antecipar a descoberta de falhas e incompatibilidades, e aumentar a segurança e a confiabilidade (LUFT, 2012).

De acordo com o CMMI/SEI (2010), "a validação tem como objetivo demonstrar que um produto ou um componente de um produto cumpra com o seu uso planejado quando colocado em um ambiente planejado para ele". Através de uma série de testes, é determinado se um software cumpre com as especificações e requisições do usuário (VILLALÓN et al., 2015).

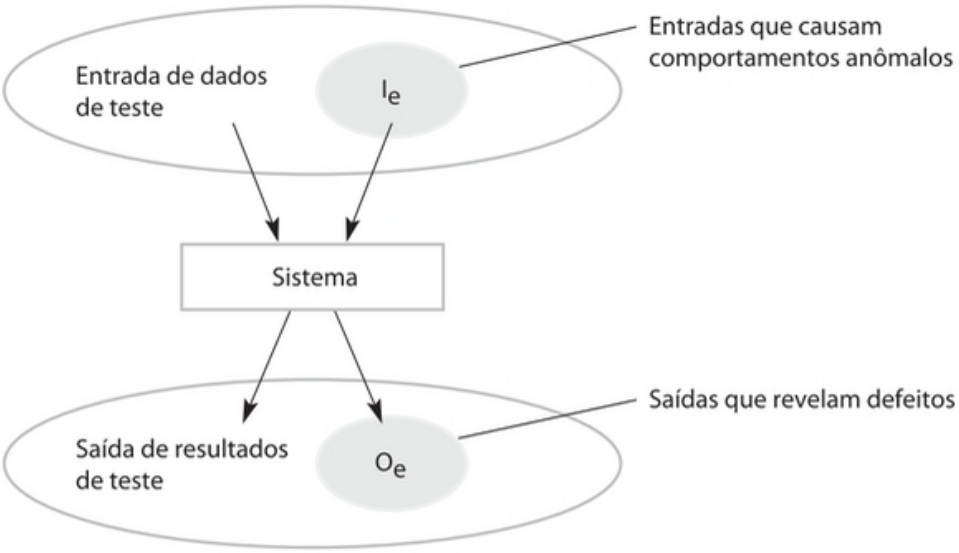
De acordo com Sommerville (2011), o processo de testes tem dois objetivos distintos:

1) Demonstrar ao desenvolvedor e ao cliente que o software atende a seus requisitos. Ou seja, para cada característica do sistema ou requisito do documento de requisitos, deve haver pelo menos um teste (testes de validação).

2) Descobrir situações em que o software se comporta de maneira incorreta, indesejável ou de forma diferente das especificações. Ou seja, procurar defeitos (bugs) no software testado (testes de defeitos).

É como pensar no sistema sendo uma caixa-preta em que aceita um conjunto de entradas. Não é necessário saber como essas entradas são processadas, é necessário apenas que essas entradas sejam processadas, e gerar um conjunto de informações de saída, ou seja, o resultado do processo de transformação. A Figura 4 demonstra as diferenças entre os testes de validação e os testes de defeitos. O sistema aceita entradas a partir de algum conjunto de entradas I e gera saídas em um conjunto de saídas O. Algumas das saídas estarão erradas, ou seja, saídas que revelam defeitos. Estas são as saídas no conjunto O_e, geradas pelo sistema em resposta a entradas definidas no conjunto I_e. A prioridade nos testes de defeitos é encontrar essas entradas definidas no conjunto I_e, pois elas revelam problemas com o sistema. Testes de validação envolvem os testes com entradas corretas que estão fora do I_e. Estes estimulam o sistema a gerar corretamente as saídas (SOMMERVILLE, 2011).

Figura 4 – Um modelo de entrada-saída de teste de programa



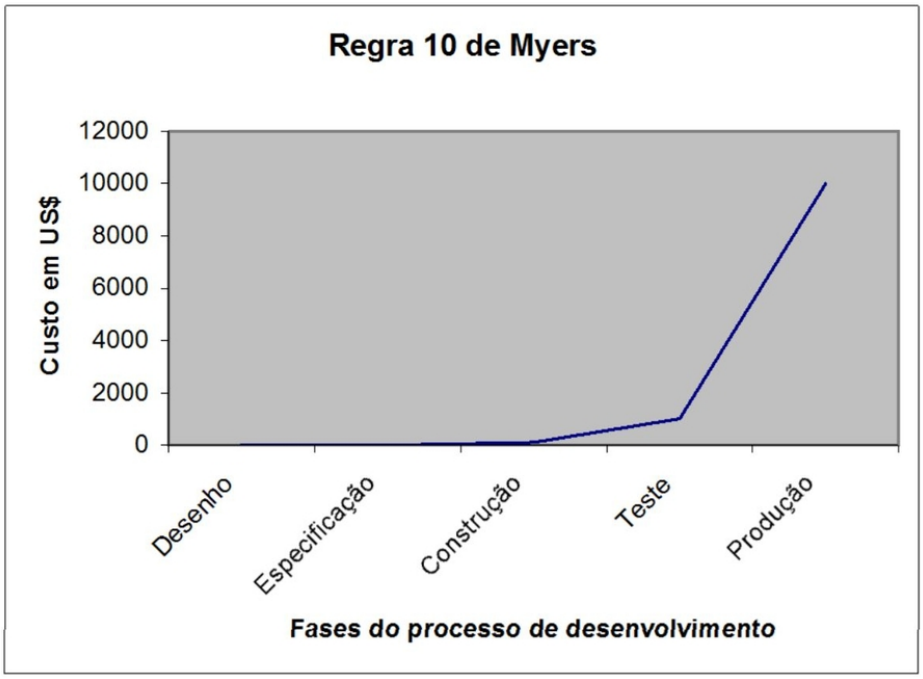
Ian Sommerville (2011)

Pressman (2011) diz que o teste é um conjunto de atividades que podem ser planejadas com antecedência e executadas sistematicamente. O teste de software é um elemento de um tópico maior, mais conhecido como verificação & validação (V&V). Muitas vezes confundidas ou usadas no mesmo sentido, verificação e validação tem significados diferentes. Verificação é conjunto de tarefas que garantem que o software foi implementado de maneira correta (“estamos criando o produto corretamente?”). Validação é o conjunto de tarefas que garantem que o software atende os requisitos definidos (“estamos criando o produto certo?”) (MPS.BR, 2016).

O objetivo principal do teste é encontrar erros antes que eles sejam entregues ao usuário final. De acordo com Myers (1979 apud BARTIÉ, 2013), o teste não tem como objetivo apenas provar uma boa funcionalidade de um determinado software, pois assim, poucos defeitos iriam ser encontrados, uma vez que as energias iriam ser direcionadas apenas para comprovar isso. Por isso o teste tem como objetivo identificar erros, pois assim, um número maior de problemas será encontrado, dado que os profissionais da área irão buscar por vários cenários para avaliar o comportamento do software.

O teste de software é importante, pois quanto mais rápido um defeito for encontrado, mais barato será para ser corrigido. De acordo com a Regra 10 de Myers (MYERS, 1979 apud BASTOS et al., 2007), a medida que vão se concluindo as fases do projeto de um software, o custo dos defeitos é multiplicado por 10. A Figura 5 demonstra quão caro pode ficar o custo de um defeito ao longo do desenvolvimento de um software.

Figura 5 – Custo de defeito



Bastos (2007)

Antes de iniciar a execução de testes, engenheiros de testes devem criar casos de teste. De acordo com a ISO/IEC/IEEE 24765 (2010), caso de teste é um conjunto de entradas, condições, e resultados esperados desenvolvido para um objetivo particular, tal como validar uma função específica do software ou verificar a conformidade com um requisito específico. A ISO/IEE/829 (2008) contém as especificações de um caso de teste, conforme demonstrado a seguir:

1) **Identificador:** Descreva o identificador exclusivo necessário para cada caso de teste para que ele possa ser diferenciado de todos os outros casos de teste.
2) **Objetivo(s):** Identifique e descreva brevemente o foco especial ou objetivo para o caso de teste.
3) **Entrada(s):** Especifique cada entrada necessária para executar cada caso de teste.
4) **Resultado(s):** Especifique todas as saídas e o comportamento esperado exigido dos itens de teste.
5) **Ambiente necessário:** Descrever o ambiente de teste necessário para configuração, execução e registro de resultados de teste:
    1) Hardware: Especifique as características e configurações do hardware necessário para executar este caso de teste.

2) Software: Especifique todas as configurações de software necessárias para executar este caso de teste. Isso pode incluir software de sistema, como sistemas operacionais, compiladores, simuladores e ferramentas de teste.
3) Outros: Especifique quaisquer outros requisitos ainda não incluídos, se houver algum.

O Quadro 1 representa um modelo simples de caso de teste onde demostra uma validação de login (requisito/característica de um sistema) em um determinado software.

Quadro 1 – Modelo de caso de teste

Caso de teste	
Identificador:	01
Objetivo(s):	Fazer login
Entrada(s):	Usuário e Senha
Resultado(s):	Login efetuado com sucesso
Ambiente necessário:	Conectado a uma rede móvel ou Wi-Fi

Elaborado pelo autor

3.3 Classificação

De acordo com o CENTRAL STATISTICS OFFICE (2018), uma classificação é um conjunto ordenado de categorias relacionadas usadas para agrupar dados de acordo com as suas semelhanças. Consiste em códigos e descritores e permite que respostas de pesquisas sejam colocadas em categorias significativas, a fim de produzir dados úteis. Ainda de acordo com o CENTRAL STATISTICS OFFICE (2018), uma classificação é uma ferramenta útil para quem desenvolve pesquisas estatísticas. É uma estrutura que simplifica o tópico em estudo e facilita a categorização de todos os dados ou respostas recebidas.

> *"A classificação é uma função importante para a transparência e para o compartilhamento de informações, as quais são caminhos para tomadas de decisões, para a preservação da memória técnica e administrativa das organizações contemporâneas e para o exercício da cidadania. É reconhecida como matricial, isto é, a partir dela é que outras funções ou outras intervenções se consolidam. Desde o momento em que seleciona o documento para guardar, o classificador trabalha com uma série de funções que discutem a dimensão do conhecimento a qual permite a prioridade de procedimentos"* (NUNES, 2007).

O ato de classificar é fundamento do ato de conhecer. De acordo com Langridge (1977), com a intenção de reconhecer o conhecimento alheio, a classificação é necessária, pois esta ocorre a todo momento, na vida do indivíduo, e é igualmente necessária

para que se proceda a uma comunicação eficiente. Com tudo, a classificação como preceito de organização está presente em toda atividade humana e é a base da interação social.

A classificação possui dois lados. O primeiro é o que contém todo o estoque de informações (acervo, arquivo, etc.). O outro lado é o do receptor de toda essa bagagem, o usuário. Desta forma, pode-se considerar que a classificação é uma intermediadora entre o sujeito e o objeto, faz a comunicação entre eles, através da organização, seja de qual suporte for. Tem papel fundamental em quase todas as áreas do conhecimento (NUNES, 2007).

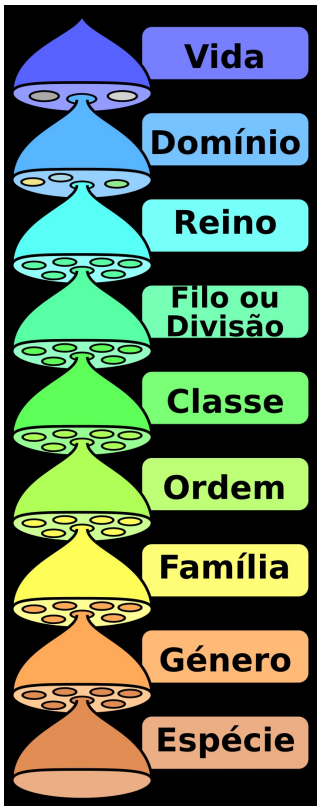
Classificar é, desse modo, dividir em grupos ou classes, segundo as diferenças e semelhanças. É assentar os conceitos, de acordo com as suas semelhanças e diferenças, em certo número de grupos metodicamente distribuídos (PIEDADE, 1983). É a ordenação de um conjunto de dados, de acordo com características que os unem ou diferem de outros grupos.

Com base nas diferentes definições de classificação mostrado nesta seção, a classificação é de suma importância pois ela irá ajudar a compreender melhor um determinado assunto, e com isso, ser utilizada para tomada de decisões futuras. Neste trabalho por exemplo, a classificação irá ajudar os profissionais da área de testes a tomar decisões dos testes a serem realizados em suas próximas execuções.

3.3.1 Taxonomia

O termo taxonomia não é novo, Linné (1735 apud LEAL, 2009) conceituou como sendo uma estrutura que permite classificar organismos vivos, produtos ou livros em grupos hierarquicamente organizados por série para facilitar a sua identificação, estudo e localização distribuindo-os em conjuntos ou classes. Ele foi responsável por criar um tipo de classificação que dividia em grupos os seres vivos, hierarquicamente, baseado em suas características em comum. É possível dizer que praticamente tudo pode ser classificado de acordo com algum esquema taxonômico (LEAL, 2009). A Figura 6 mostra de forma simplificada a taxonomia criada por Karl Von Linné.

Figura 6 – A hierarquia da classificação científica dos seres vivos



Peter Halasz (2007)

A NATIONAL INFORMATION STANDARDS ORGANIZATION (NISO) (2005) define taxonomia como sendo *"Uma coleção de termos de vocabulário controlado organizados em uma estrutura hierárquica. Cada termo em uma taxonomia está em um ou mais relacionamentos pai/filho (mais amplo/mais estreito) com outros termos da taxonomia".* As taxonomias buscam por organizar o conhecimento e/ou a informação em relações hierárquicas entre os seus termos. Contudo, essa organização ajuda a conciliar a busca mais precisa de determinado assunto no momento em que o mesmo é recuperado.

De acordo com Terra et al. (2004), "A taxonomia é um sistema para classificar e facilitar o acesso à informação, e que tem como objetivos: representar conceitos através de termos; agilizar a comunicação entre especialistas e outros públicos; encontrar o consenso; propor formas de controle da diversidade de significação; e oferecer um mapa de área que servirá como guia em processos de conhecimento". Logo, e taxonomia é um vocabulário controlado de uma determinada área do conhecimento, e acima de tudo um instrumento que permite comunicar, recuperar e alocar informações dentro de um sistema, de maneira lógica.

As taxonomias não têm um modelo pronto que servirá para todos os ambientes e nem possuem uma tecnologia única. O seu entendimento é subjetivo, aplicável e

moldado à realidade situacional no qual ela está sendo inserida. Ou seja, a forma que a taxonomia será utilizada, depende do contexto o qual a organização irá implementar (SIQUEIRA et al., 2014).

De acordo com Novo (2010), “A estrutura taxonômica representa através de conceitos o mapeamento de um domínio de conhecimento e pode ser apresentado por ”mapas conceituais“. Embora assumam geralmente formas hierárquicas. As taxonomias atualmente são apresentadas sob novas formas visuais, como, por exemplo as árvores hiperbólicas”.

3.4 Considerações finais

Este capítulo apresentou referências que servem como base conceitual e auxiliam o desenvolvimento deste trabalho. Foram apresentados conceitos importantes sobre qualidade de software, teste de software, classificação e taxonomia. Assim como também foi mostrado definições abordadas por diferentes autores.

4 TRABALHOS RELACIONADOS

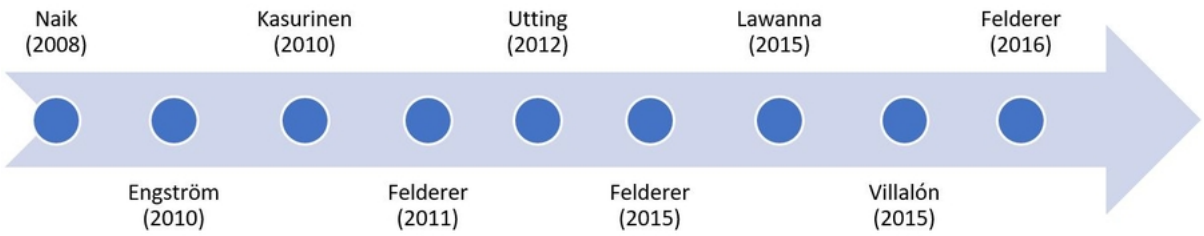
Neste capítulo é mostrado uma análise dos trabalhos que tem alguma relação com essa pesquisa, e que também serviram como auxílio para o desenvolvimento deste trabalho.

4.1 Visão geral

Foi feito uma busca em diversas bibliotecas digitais com o intuito de encontrar por trabalhos que tivessem alguma relação com o tema desta pesquisa. As principais bibliotecas digitais utilizadas para encontrar estes trabalhos, foram: (1) IEEE Xplore Digital Library, (2) Google Acadêmico, (3) Springer Link, (4) Science Direct e (5) ACM Digital Library.

A partir desta busca na literatura, foi possível encontrar trabalhos relacionados a classificação ou seleção de testes. Esses trabalhos têm como objetivo apresentar diferentes formas de classificar e/ou selecionar testes de software com o intuito de apoiar a seleção de testes mais adequados ao escopo do projeto e melhorar o desempenho da execução de testes. A Figura 7 mostra uma linha do tempo da data de publicação destes trabalhos relacionados.

Figura 7 – Linha do tempo dos trabalhos relacionados



Elaborado pelo autor

4.2 Análise dos trabalhos

Naik e Tripathy (2008) desenvolveram uma taxonomia de testes de sistema com o intuito de ajudar os engenheiros de testes. Eles explicam que o objetivo do teste do sistema é estabelecer se uma implementação está de acordo com os requisitos especificados pelos clientes. É preciso muito esforço para garantir que os requisitos do cliente sejam atendidos e o sistema seja aceitável. Uma variedade de testes é executada para atender a uma ampla gama de expectativas não especificadas também. Com isso, eles

desenvolveram essa taxonomia com o intuito de ter um melhor conhecimento sobre o assunto e trazer as seguintes vantagens para os engenheiros de testes:

- Priorizar suas tarefas com base nas categorias de testes.
- O planejamento da fase de testes do sistema com base na taxonomia de testes permite que um engenheiro de testes obtenha um conjunto de testes bem balanceados.
- Focar com precisão os diferentes aspectos de um sistema, um de cada vez, enquanto avaliam sua qualidade.

Engström, Runeson e Skoglund (2010) conduziram uma revisão sistemática das avaliações empíricas das técnicas de seleção de testes de regressão. Foram identificados 27 artigos reportando 36 estudos empíricos, 21 experimentos e 15 casos de estudo. No total, 28 técnicas de seleção de testes de regressão foram avaliadas. Eles então apresentaram uma análise qualitativa de suas descobertas, uma visão geral das técnicas de seleção de testes de regressão e uma evidência empírica relacionada. Por fim, eles identificaram alguns problemas básicos no campo de testes de regressão e acreditam que seu trabalho possa ser de grande valor para futuras pesquisas na área.

Kasurinen, Taipale e Smolander (2010) fizeram uma classificação sobre como as organizações de software decidem quais casos de teste selecionar para os seus projetos. Este estudo examinou como casos de teste são projetados e selecionados nos planos de testes em 12 empresas de software. Um questionário foi enviado para 31 empresas de software e 36 profissionais das 12 principais empresas foram entrevistados. Os métodos que as empresas usam foram divididos em 6 categoria. Por fim, foi relatado que os métodos que as empresas de software usam para selecionar os casos de teste tem foco em duas abordagens: seleção com base de risco e seleção com base no design. A primeira tem foco em testar as partes do software onde será mais custoso de ser corrigido uma vez que o software já foi lançado. A segunda tem foco em testar e garantir que o software é capaz de cumprir com as principais funcionalidades as quais ele foi projetado. O objetivo principal deste estudo foi entender e explicar as estratégias de empresas de software na escolha de seus casos de teste.

Felderer, Agreiter e Zech (2011) propuseram uma classificação para testes de segurança baseado em modelo (MBST). Testes de segurança são diferentes dos testes tradicionais, o foco deles é garantir que determinada aplicação está livre de problemas de segurança. O teste de segurança procura por falhas que possam exibir ou modificar dados sensíveis. A classificação desenvolvida pelos autores permite entender melhor quais áreas do MBST estão bem definidas tanto em pesquisa quanto na prática e servir

como orientação para engenheiros de testes decidirem qual abordagem melhor se encaixa em determinadas circunstâncias. Os autores mostraram uma visão geral das diferentes abordagens existentes na literatura sobre testes de segurança. A partir dessa abordagem inicial, eles então dividiram essas abordagens em diferentes categorias, e assim, desenvolveram a classificação. A intenção do estudo deles é servir como base para futuros trabalhos assim como diminuir a necessidade de um alto nível de conhecimento para realizar testes de segurança.

Utting, Pretschner e Legeard (2012) desenvolveram uma taxonomia que cobre os aspectos chaves das abordagens de teste baseado em modelo (MBT). Foi realizado também uma classificação de diferentes ferramentas de MBT a partir da taxonomia desenvolvida. A intenção deles é ajudar com o entendimento das características, similaridades e diferenças sobre essas abordagens. A partir de conceitos fundamentais da MBT e suas terminologias, a taxonomia dividiu as abordagens de MBT em seis diferentes dimensões. A taxonomia desenvolvida por eles fornece as características essenciais de várias abordagens da MBT, tanto acadêmicas quanto industriais. A utilidade da taxonomia foi demonstrada através da classificação de várias ferramentas MBT.

Felderer e Fourneret (2015) realizaram uma classificação sistemática das abordagens de testes de segurança de regressão disponíveis. Eles extraíram abordagens a partir de bibliotecas digitais relevantes (ACM Digital Library, IEEE Xplore, Sciente Direct, SpringerLink, entre outras) com base em uma rigorosa estratégia de busca e seleção. Eles forneceram então uma classificação dessas abordagens por meio de diferentes tipos de critérios. No total, 17 testes de segurança de regressão foram classificados a partir de 18 artigos selecionados. Os resultados mostraram que o campo de testes de segurança de regressão é relativamente novo e que as abordagens baseadas em modelos são dominantes neste tipo de teste. Com base nos resultados, eles identificaram vários potenciais para futuras pesquisas.

Lawanna (2015) propôs um algoritmo chamado “Software Testing Improvement” (STI). Este algoritimo foi desenvolvido com o propósito de melhorar a habilidade em escolher casos de teste em comparação a três diferentes métodos existentes: RA (Retest-All Method), RD (Random Technique) e ST (Safe Test Technique). A técnica RA retesta todos os casos de teste, o que demanda muito custo e tempo. A técnica RD faz uma escolha aleatória de casos de teste, o que pode acabar escolhendo casos de teste ineficientes. A terceira técnica, ST, faz uma busca por casos de teste que produziram erros em versões anteriores, entretanto, está técnica pode deixar escapar casos de teste que deveriam ser escolhidos por causa de novos erros que foram introduzidos. O algoritmo STI faz uma análise mais profunda e tem os seguintes passos: Encontrar funções modificadas; determinar linhas de códigos modificadas;

encontrar o número apropriado de casos de teste; selecionar casos de teste. Para validar o método proposto, sete programas desenvolvidos por Siemen Suite foram usados nesta pesquisa. Entre os quatro métodos avaliados (RA, RD, ST e STI), o STI foi o que teve melhor eficiência.

Villalón et al. (2015) propuseram estabelecer uma taxonomia de projeto de software que permite agrupar projetos com base em testes comuns para projeto de software. A taxonomia que eles desenvolveram tem foco no estabelecimento de projetos com base no método proposto por Bayona-Oré et al. (2014), que é composto por 5 fases: Planejamento, identificação e construção da taxonomia, projeção e construção da taxonomia, teste e validação, e implantação da taxonomia. Foi feita então uma pesquisa para procurar por palavras chaves que definem os principais tipos de projetos nas empresas de software e suas subcategorias. Por fim, eles desenvolveram um questionário a profissionais de TI para validar a taxonomia criada e tiverem uma boa aprovação.

Felderer et al. (2016) forneceram uma taxonomia para abordagens de testes de segurança baseado em modelo. Ao todo, 119 publicações sobre testes de segurança baseado em modelo foram extraídas das cinco mais relevantes bibliotecas digitais e foram usadas para a criação da taxonomia. O propósito principal deste trabalho foi fornecer um esquema de classificação para abordagens de testes de segurança, um importante ponto de partida para pesquisas futuras sobre testes de segurança baseados em modelos e ajudar a esclarecer os principais problemas do campo e a mostrar possíveis alternativas e direções.

4.3 Considerações finais

Os trabalhos mencionados acima têm foco em classificar ou selecionar diferentes tipos de testes, tais como teste de segurança e regressão. Neste trabalho, o foco é a análise e classificação de casos de teste. Apesar de alguns trabalhos não terem relação direta com esta pesquisa, a metodologia utilizada por eles serve como guia.

5 TAXONOMIA DE CASOS DE TESTE DE SOFTWARE DE CAIXA PRETA

Este capítulo tem como objetivo demonstrar a proposta deste trabalho, uma taxonomia de casos de teste de software de caixa preta. A taxonomia foi formada com base em diferentes fontes de pesquisa que também foram utilizadas na revisão bibliográfica e trabalhos relacionados.

5.1 Visão geral

De acordo com Terra et al. (2004), um dos objetivos da taxonomia é classificar a informação, de uma forma hierárquica, de maneira que seja facilitado o acesso a ela, melhorando a comunicação entre os principais usuários, tanto especialistas, quanto entre um público qualquer.

O modelo da taxonomia teve como base a proposta de Bayona-Oré et al. (2014) que é composta pelas seguintes fases:

1) Planejamento
    - Identificar a área de estudo
    - Definir os objetivos da taxonomia
    - Definir o escopo da taxonomia proposta
2) Identificação e extração de informação
    - Identificar as fontes de informação
    - Extrair todos os termos e identificar as categorias candidatas
3) Concepção e construção da taxonomia
    - Verificar a lista de termos e definir os critérios
    - Definir o primeiro nível da taxonomia
    - Definir os níveis subsequentes da taxonomia
    - Validar a taxonomia com *stakeholders* e/ou especialistas

Na etapa de identificação e extração de informação, vários termos foram encontrados. Com isso, esses termos passaram pelos seguintes critérios para que pudessem ser selecionados como categorias candidatas:

1) A definição indica que é um teste de caixa preta

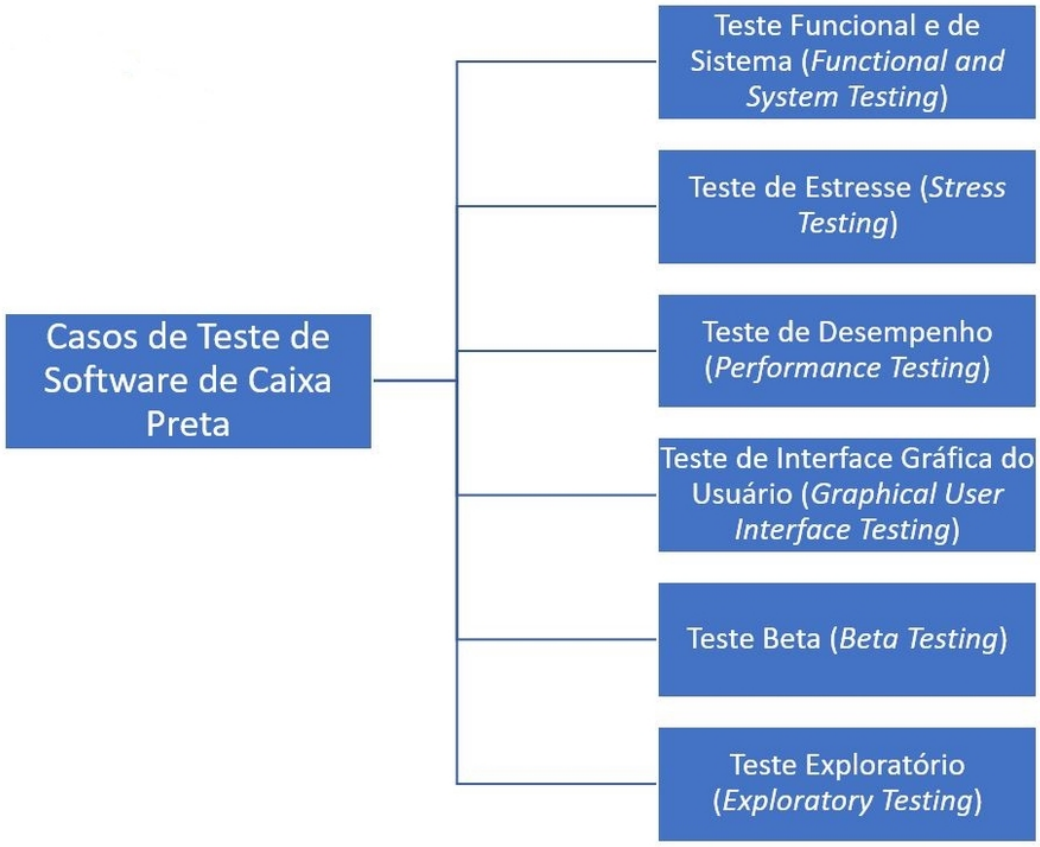
2) Não existe outro termo já selecionado com definição semelhante

A estrutura da taxonomia é mostrada nas próximas seções, ela foi montada a partir de estudos de propostas de autores como Tuteja e Dubey (2012), Nidhra e Dondeti (2012), Naik e Tripathy (2008), Itkonen, Mäntylä e Lassenius (2009) e Jorgensen (2014). A estrutura não tem a intenção de esgotar as possibilidades apresentadas entre os diversos tipos de testes de caixa preta, mas serve como um direcionamento para o pensamento sistematizado a respeito do domínio aqui apresentado. A visão completa da taxonomia pode ser encontrada no Apêndice A.

5.2 Primeiro e segundo nível

A Figura 8 mostra o primeiro e segundo nível da taxonomia. No topo da hierarquia (primeiro nível) é apresentado o assunto base da taxonomia, casos de teste de software de caixa preta. Fazem parte desta base central (segundo nível) os tipos de testes sugeridos por Tuteja e Dubey (2012) e Nidhra e Dondeti (2012). A decomposição do restante da taxonomia é apresentada nas próximas subseções deste capítulo.

Figura 8 – Taxonomia de casos de teste de software de caixa preta



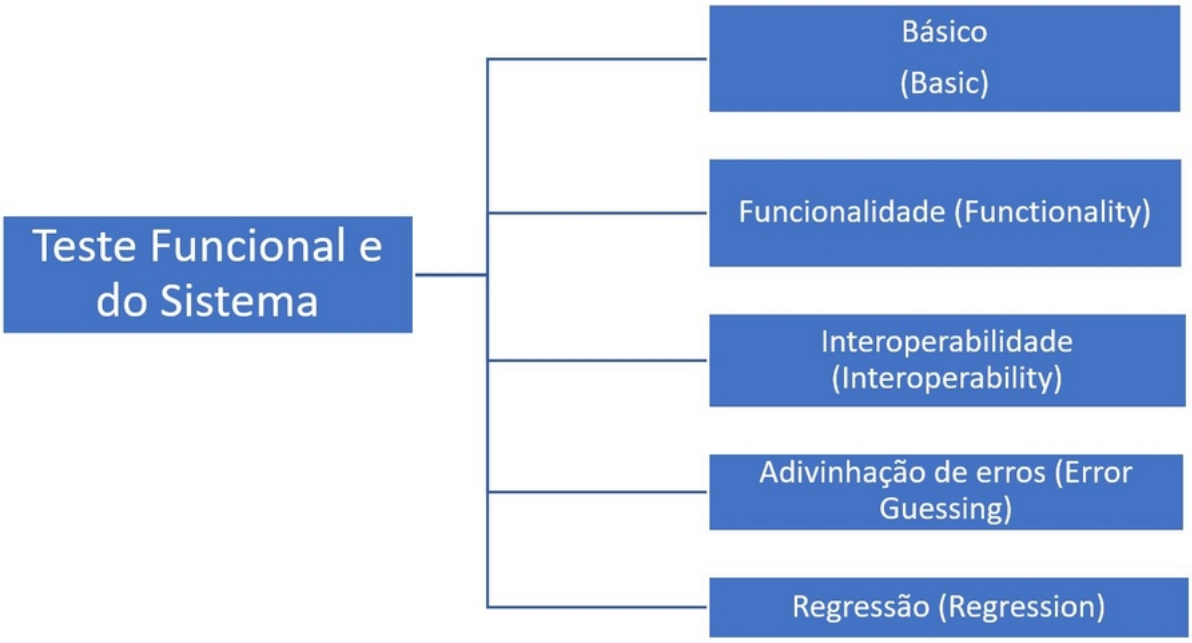
Elaborado pelo Autor

5.3 Nível subsequente

5.3.1 Funcional e de Sistema

Funcional são os testes conduzidos para avaliar a conformidade de um sistema ou componente com requisitos funcionais especificados. Sistema são os testes conduzidos em um sistema completo e integrado com o intuito de avaliar a conformidade do sistema com seus requisitos especificados (INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, 1990).

Figura 9 – Taxonomia de teste funcional e de sistema



Elaborado pelo Autor

Para sua decomposição, foram sugeridos os seguintes componentes: Básico *(Basic)*, Funcionalidade (*Functionality)*, Robustez (Robustness), Interoperabilidade *(Interoperability)*, Escalabilidade (*Scalability)*, Matrizes Ortogonais (*Pairwise)*, Aleatoriedade (*Random)*, Adivinhação de erros (*Error Guessing)* e Regressão (*Regression)* (NAIK; TRIPATHY, 2008) como demonstrado na Figura 9.

5.3.1.1 Básico

Os testes básicos fornecem testes limitados do sistema em relação aos principais recursos em uma especificação de requisitos. São realizados para garantir que

as funções comumente usadas funcionem para nossa satisfação (NAIK; TRIPATHY, 2008).

5.3.1.2 Funcionalidade

Os testes de funcionalidade verificam o sistema o mais completo possível em toda a faixa de requisitos especificada no documento de especificação de requisitos (NAIK; TRIPATHY, 2008).

5.3.1.3 Interoperabilidade

Os testes de interoperabilidade são projetados para verificar a capacidade do sistema de interoperar com produtos de terceiros. Um teste de interoperabilidade normalmente combina diferentes elementos de rede em um ambiente de teste para garantir que eles funcionem juntos. Em outras palavras, os testes são projetados para garantir que o software possa ser conectado a outros sistemas e operadores(NAIK; TRIPATHY, 2008).

5.3.1.4 Adivinhação de erros

A adivinhação de erro é uma técnica de design de caso de teste em que um engenheiro de teste usa a experiência para adivinhar os tipos e locais prováveis de defeitos e projetar testes especificamente para revelar os defeitos. Embora a experiência seja muito útil para adivinhar erros, é útil adicionar alguma estrutura à técnica. É bom preparar uma lista de tipos de erros que podem ser descobertos. A lista de erros pode nos ajudar a adivinhar onde os erros podem ocorrer. Essa lista deve ser mantida a partir da experiência adquirida em projetos de testes anteriores (NAIK; TRIPATHY, 2008).

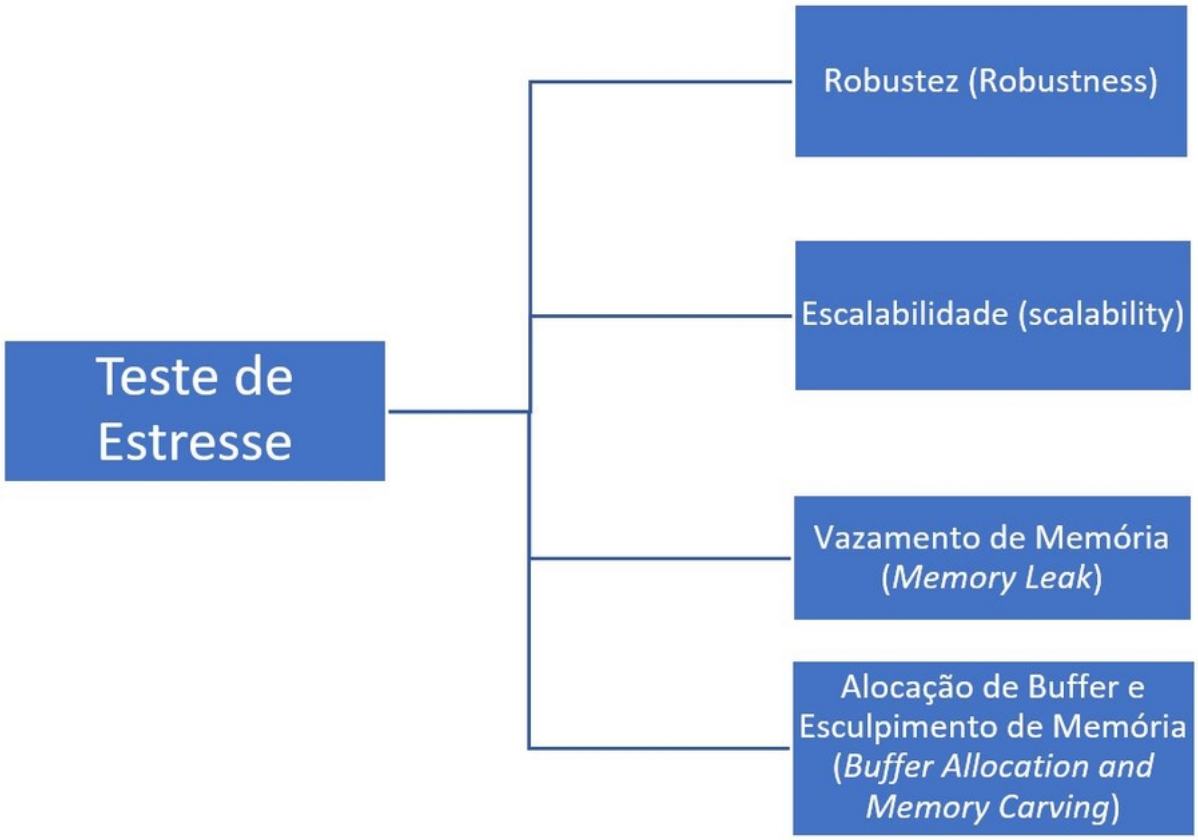
5.3.1.5 Regressão

Nesta categoria, novos testes não são projetados. Em vez disso, os casos de teste são selecionados do conjunto existente e executados para garantir que nada seja quebrado na nova versão do software. A ideia principal do teste de regressão é verificar se nenhum defeito foi introduzido na parte inalterada de um sistema devido às alterações feitas em outras partes do sistema. Durante o teste do sistema, muitos defeitos são revelados e o código é modificado para corrigir esses defeitos. Como resultado da modificação do código, outros defeitos podem ocorrer (NAIK; TRIPATHY, 2008).

5.3.2 Estresse

Estresse são os testes conduzidos para avaliar um sistema ou componente nos limites de seus requisitos especificados ou além deles (INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, 1990).

Figura 10 – Taxonomia de teste de estresse



Elaborado pelo Autor

Para sua decomposição, foram sugeridos os seguintes testes: Vazamento de Memória (*Memory Leak*) e Alocação de Buffer e Gravação de Memória (*Buffer allocation and memory carving*) (NAIK; TRIPATHY, 2008) como demonstrado na Figura 10.

5.3.2.1 Robustez

Os testes de robustez checam quão sensível é um sistema a entradas erradas e mudanças em seu ambiente operacional. O propósito é fazer o sistema falhar deliberadamente, não como um fim em si mesmo, mas como um meio para encontrar erros (NAIK; TRIPATHY, 2008).

5.3.2.2 Escalabilidade

Os testes de escalabilidade são projetados para verificar se o sistema pode aumentar os seus limites de engenharia. Um sistema pode funcionar em um cenário de uso limitado, mas pode não aumentar de escala. A ideia é testar o limite do sistema, isto é, a magnitude da demanda que pode ser colocada no sistema enquanto continua a atender aos requisitos de latência e taxa de transferência (NAIK; TRIPATHY, 2008).

5.3.2.3 Vazamento de Memória

Um vazamento de memória é a perda de memória disponível quando um programa não retorna a memória obtida para uso temporário (NAIK; TRIPATHY, 2008). Este é um problema bem conhecido de software, e encontrar ele é o objetivo deste tipo de teste.

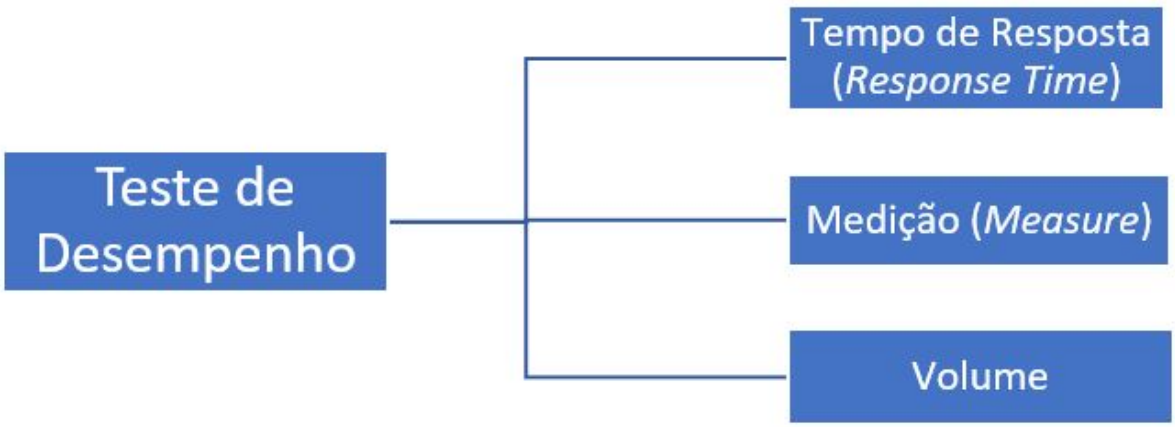
5.3.2.4 Alocação de Buffer e Gravação de Memória

O problema de alocação de buffer significa a má distribuição de espaços de buffer entre os nós de uma rede (ANANTHARAM, 1989). Esculpimento de memória é o processo de remontar a memória do computador a partir de fragmentos na ausência de metadados do sistema de arquivos. Analisar quão bem um software realiza a alocação de buffer e gravação de memória é o objetivo deste tipo de teste.

5.3.3 Desempenho

Os testes de desempenho são projetados para determinar o desempenho do sistema real em comparação ao esperado. As métricas de desempenho que precisam ser medidas variam de aplicativo para aplicativo (NAIK; TRIPATHY, 2008).

Figura 11 – Taxonomia de teste de desempenho



Elaborado pelo Autor

Para sua decomposição, foram sugeridos os seguintes testes: Tempo de Resposta (*Reponse Time)*, Medição *(Measure)* e Volume (NAIK; TRIPATHY, 2008) como demonstrado na Figura 11.

5.3.3.1 Tempo de Resposta

Testes para calcular o tempo de resposta de uma requisição para a resposta. Um exemplo é o seguinte: o tempo de resposta deve ser menos de 1 milissegundo em 90% do tempo em uma aplicação “pressione para falar” *(push-to-talk)* (NAIK; TRIPATHY, 2008).

5.3.3.2 Medição

Testes realizados com o intuito de fazer alguma medição com resultados pré-estabelecidos. Como por exemplo, medir a quantidade de tempo que a bateria de um dispositivo móvel aguenta em uma ligação (NAIK; TRIPATHY, 2008).

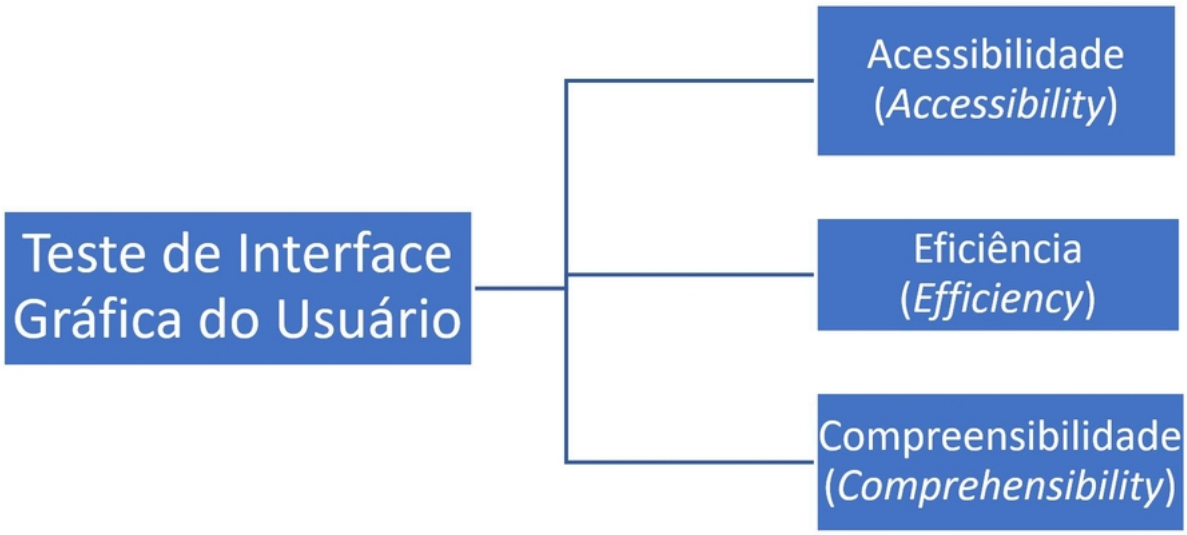
5.3.3.3 Volume

O teste de volume é um teste não funcional que se refere ao teste de um aplicativo de software com uma grande quantidade de dados a serem processados para verificar a eficiência do aplicativo. O principal objetivo desse teste é monitorar o desempenho do aplicativo em vários volumes de banco de dados (NAIK; TRIPATHY, 2008).

5.3.4 Interface Gráfica do Usuário

Testes de Interface Gráfica do Usuário (usabilidade) são os testes conduzidos para descobrir informações de como a interface do usuário do sistema corresponde a maneira natural humana de pensar e agir (KAIKKONEN et al., 2005).

Figura 12 – Taxonomia de teste de interface gráfica do usuário



Elaborado pelo Autor

Para sua decomposição, foram sugeridos os seguintes testes: Acessibilidade *(Accessibility)*, Capacidade de Resposta *(Responsiveness)*, Eficiência *(Efficiency)* e Compreensibilidade *(Comprehensibility)* (NAIK; TRIPATHY, 2008) como demonstrado na Figura 12.

5.3.4.1 Acessibilidade

Testes para verificar se o usuário pode fazer o que quer e quando quiser de uma maneira clara. Inclui fatores ergonômicos como cor, forma, som e tamanho da fonte (NAIK; TRIPATHY, 2008).

5.3.4.2 Eficiência

Testes para verificar se o usuário consegue fazer o que desejam (usar alguma funcionalidade) com um número mínimo de etapas e tempo (NAIK; TRIPATHY, 2008).

5.3.4.3 Compreensibilidade

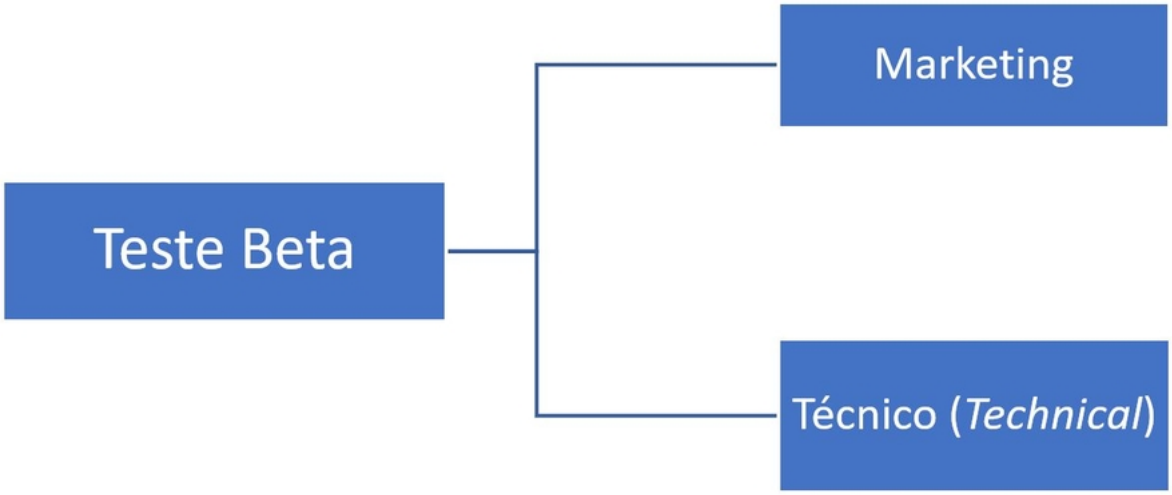
Testes para verificar se o usuário consegue entender a estrutura do software com um mínimo de esforço (NAIK; TRIPATHY, 2008).

5.3.5 Beta

Beta são os testes realizados por clientes selecionados, com o intuito de receber um feedback de possíveis melhorias antes de lançar para o público geral (NIDHRA;

DONDETI, 2012).

Figura 13 – Taxonomia de teste beta



Elaborado pelo Autor

Para sua decomposição, foram sugeridos os seguintes testes: Marketing e Técnico *(Technical)* (NAIK; TRIPATHY, 2008) como demonstrado na Figura 13.

5.3.5.1 Marketing

O objetivo aqui é criar conscientização e interesse no produto entre os potenciais compradores (NAIK; TRIPATHY, 2008).

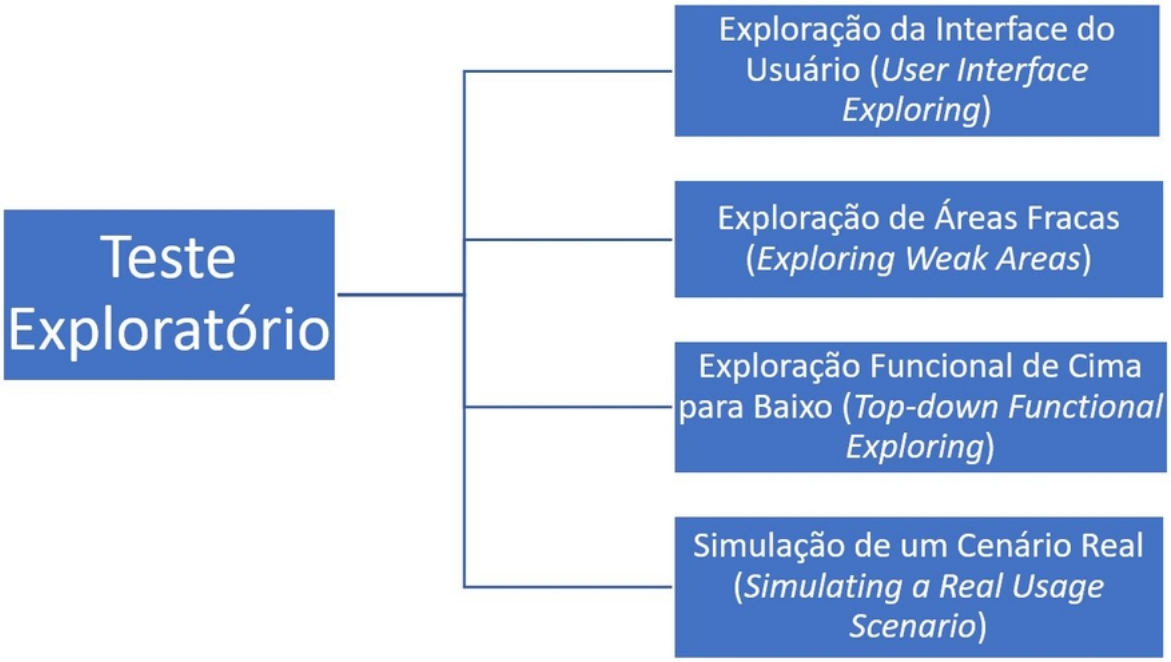
5.3.5.2 Técnico

O objetivo aqui é obter feedback sobre a usabilidade do produto, em um ambiente real com diferentes configurações, de um pequeno número de clientes amigáveis. A ideia é obter feedback de um número limitado de clientes que dedicam uma quantidade considerável de tempo e pensam na sua avaliação (NAIK; TRIPATHY, 2008).

5.3.6 Exploratório

Exploratório são os testes considerados de extensão, os quais não tem pré-requisitos a não ser o próprio sistema a ser testado. O testador controla o projeto dos testes, e a medida em que ele vai ganhando mais informações, ele vai usando essas informações para projetar novos e melhores testes (BACH, 2003).

Figura 14 – Taxonomia de teste exploratório



Elaborado pelo Autor

Para sua decomposição, foram sugeridos os seguintes testes: Exploração da Interface do Usuário *(User Interface Exploring),* Exploração de Áreas Fracas *(Exploring Weak Areas),* Orientado a Aspectos *(Aspect Oriented)*, Exploração Funcional de Cima para Baixo *(Top-down Functional Exploring),* Simulação de um Cenário Real (Simulating a Real Usage Scenario) e Teste de Fumaça por Intuição e Experiência *(Smoke Testing By Intuition and Experience)* (ITKONEN; MÄNTYLÄ; LASSENIUS, 2009) como demonstrado na Figura 14.

5.3.6.1 Exploração da Interface do Usuário

O testador estrutura os testes a partir da interface do usuário e segue de funcionalidade por funcionalidade para cobrir toda a interface do usuário (ITKONEN; MÄNTYLÄ; LASSENIUS, 2009).

5.3.6.2 Exploração das Áreas Fracas

O testador se preocupa em analisar áreas que tem um maior potencial de ocorrer erros, baseado em experiências anteriores (ITKONEN; MÄNTYLÄ; LASSENIUS, 2009).

5.3.6.3 Exploração Funcional de Cima para Baixo

O objetivo aqui é obter primeiro um alto nível de compreensão da função e, em seguida, aprofundar a confiança em sua qualidade, passo a passo (ITKONEN, 2011).

5.3.6.4 Simulação de um Cenário Real

Testes com o intuito de simular um cenário real (como o nome já propõe) e ter uma validação mais perto de como seria com o usuário final (ITKONEN; MÄNTYLÄ; LASSENIUS, 2009).

5.4 Considerações finais

Neste capítulo foi demonstrado a proposta deste trabalho de pesquisa. Foi mostrado cada detalhe da taxonomia. No próximo capítulo será mostrado a validação da taxonomia, que foi realizada por meio da técnica de grupo focal.

6 GRUPO FOCAL

Este capítulo tem como objetivo demonstrar o planejamento, execução e análise dos dados obtidos do grupo focal, assim como também destacar as limitações e ameaças à validade desta técnica.

6.1 Planejamento

O problema de pesquisa do grupo focal é o mesmo problema que segue nesta pesquisa, como classificar os casos de teste com o intuito de apoiar a busca e seleção de testes mais adequados ao escopo do projeto. A partir deste problema, o grupo focal tem como objetivo avaliar a taxonomia de casos de teste de software de caixa preta sob os aspectos de **completude**, **clareza** e **adequação** da estrutura proposta.

Quadro 2 – Roteiro para realização do grupo focal

Atividade	Duração	Passos
1	10 min	Apresentação dos objetivos para realização do grupo focal Apresentação da visão geral da proposta
2	10 min	O moderador apresentou o **primeiro nível da taxonomia**, e em seguida, iniciou a seguinte discussão: ● Está claro a decomposição do primeiro nível da taxonomia? ● É necessário adicionar ou remover alguma classificação?
3	10 min	O moderador apresentou o nível subsequente sobre **teste funcional e de sistema**, e em seguida, iniciou a seguinte discussão: ● Está claro a decomposição deste nível subsequente? ● É necessário adicionar ou remover alguma classificação?
4	10 min	O moderador apresentou o nível subsequente sobre **teste de estresse**, e em seguida, iniciou a seguinte discussão: ● Está claro a decomposição deste nível subsequente? ● É necessário adicionar ou remover alguma classificação?
5	10 min	O moderador apresentou o nível subsequente sobre **teste de desempenho**, e em seguida, iniciou a seguinte discussão: ● Está claro a decomposição deste nível subsequente? ● É necessário adicionar ou remover alguma classificação?
6	10 min	O moderador apresentou o nível subsequente sobre **teste de interface gráfica do usuário**, e em seguida, iniciou a seguinte discussão: ● Está claro a decomposição deste nível subsequente? ● É necessário adicionar ou remover alguma classificação?
7	10 min	O moderador apresentou o nível subsequente sobre **teste de aceitação**, e em seguida, iniciou a seguinte discussão: ● Está claro a decomposição deste nível subsequente? ● É necessário adicionar ou remover alguma classificação?
8	10 min	O moderador apresentou o nível subsequente sobre **teste beta**, e em seguida, iniciou a seguinte discussão: ● Está claro a decomposição deste nível subsequente? ● É necessário adicionar ou remover alguma classificação?
9	10 min	O moderador apresentou o nível subsequente sobre **teste exploratório**, e em seguida, iniciou a seguinte discussão: ● Está claro a decomposição deste nível subsequente? ● É necessário adicionar ou remover alguma classificação?
10	10 min	O moderador apresentou o nível subsequente sobre **teste de fronteira**, e em seguida, iniciou a seguinte discussão: ● Está claro a decomposição deste nível subsequente? ● É necessário adicionar ou remover alguma classificação?
Tempo total:	100 min	

Elaborado pelo autor

Com o intuito de alcançar o objetivo do grupo focal, foi planejado o roteiro demonstrado no Quadro 2.

Quadro 3 – Especialistas selecionados para o grupo focal

Especialista 1 (E1)	
Nível de experiência	3 anos de experiência em prática em teste de software
Dados demográficos	Idade: 23 anos Gênero: Masculino Formação: Graduado em Engenharia da Computação Função: Técnico em Teste de Software
Interesses individuais	Área de automação em testes de software
Habilidades técnicas	Java, C, Scrum, Testes
Especialista 2 (E2)	
Nível de experiência	3 anos de experiência em prática em teste de software
Dados demográficos	Idade: 27 anos Gênero: Masculino Formação: Graduado em Engenharia da Computação Função: Engenheiro de Teste de Software
Interesses individuais	Análise e requisitos de criação de use case
Habilidades técnicas	Java, Scrum
Especialista 3 (E3)	
Nível de experiência	2 anos de experiência em prática em teste de software
Dados demográficos	Idade: 26 anos Gênero: Masculino Formação: Graduado em Ciência da Computação Função: Engenheiro de Teste de Software
Interesses individuais	Automação de testes, análise de qualidade
Habilidades técnicas	Java, Android, Python, R, MatLab
Especialista 4 (E4)	
Nível de experiência	4,5 anos de experiência em prática em teste de software
Dados demográficos	Idade: 30 anos Gênero: Masculino Formação: Graduado em Engenharia Eletrônica Função: Engenheiro de Teste de Software
Interesses individuais	Interesse na área de Engenharia de QA
Habilidades técnicas	Java, C, GIT, UIAutomator, Android, JIRA, Jenkins, Gerrit
Especialista 5 (E5)	
Nível de experiência	3,5 anos de experiência em prática em teste de software
Dados demográficos	Idade: 27 anos Gênero: Masculino Formação: Graduado em Ciência da Computação Função: Engenheiro de Teste de Software
Interesses individuais	Área de automação de teste e teste de regressão
Habilidades técnicas	Java, Python, MatLab
Especialista 6 (E6)	
Nível de experiência	2 anos de experiência em prática em teste de software
Dados demográficos	Idade: 23 anos Gênero: Masculino Formação: Graduado em Ciência da Computação Função: Técnico em Teste de Software
Interesses individuais	Automação de testes e ferramentas de geração de planos de teste
Habilidades técnicas	Java, Android, Python

Elaborado pelo autor

Os especialistas foram selecionados a partir do perfil e conhecimento na área de teste de software. Foram convidados 7 participantes, e desses, 6 compareceram

a reunião. O Quadro 3 descreve o perfil destes participantes, e seus nomes foram preservados por questões de confidencialidade.

6.2 Execução

O grupo focal foi realizado no dia 12 de dezembro de 2018, com duração aproximada de 100 minutos, sendo registrado em áudio, por meio da permissão dos participantes, além das anotações no computador realizadas pelo moderador, sempre que achou necessário.

A seção teve início com a apresentação, aos participantes, do método do grupo focal, onde foi indicado o objetivo desta técnica na aplicação do contexto desta pesquisa. Foi explicado que a intenção era coletar o máximo possível de feedback (respeitando o tempo previamente estabelecido) a partir da interação entre os participantes (KITZINGER, 1994). Neste momento, foi apresentado o roteiro a ser seguido, conforme descrito no Quadro 2. Foi apresentado também a visão geral da proposta, descrevendo o problema, objetivos da pesquisa, assim como uma descrição do que é uma taxonomia. Por fim, foi solicitado aos participantes que eles se sentissem à vontade para contribuir com qualquer tipo de sugestão, opinião ou crítica sobre os temas abordados.

Foi apresentado então, aos participantes, toda a decomposição da taxonomia, passando pelo primeiro nível e por todas as categorias do nível subsequente. Para cada categoria, foi pedido que os participantes debatessem entre si e contribuíssem com as suas ideias.

6.3 Análise dos dados

Para a análise dos dados, foi realizado uma análise qualitativa, que tem como objetivo consolidar, reduzir e interpretar dados obtidos por diferentes fontes e entendê-los (SANTOS et al., 2016). Os dados qualitativos foram analisados a partir da transcrição dos áudios, que foi realizada por meio da ferramenta Microsoft Excel.

Com isso, os dados consolidados foram estruturados em tópicos com base nas atividades planejadas conforme apresentado no Quadro 2.

6.3.1 Primeiro nível da taxonomia

Com relação ao **primeiro nível da taxonomia**, todos os especialistas concordaram que a sua decomposição estava clara. Além disso, não acharam necessário adicionar ou remover nenhuma classificação. Uma vez que essa taxonomia classifica diversos tipos de teste de caixa preta, pode-se notar que alguns especialistas desco-

nheciam alguns dos tipos de testes de caixa preta proposto. Os comentários a seguir relatam essa análise:

> [. . .] Está bem claro o primeiro nível da taxonomia. E1

> [. . .] Tem mais testes do que eu conheço ou que já trabalhei. E5

6.3.2 Teste funcional e de sistema

Se tratando da classificação de **teste funcional e de sistema**, todos os especialistas concordaram que a sua decomposição estava clara e de fácil entendimento. Também não adicionariam nem removeriam nenhuma das classificações. As transcrições relatam isso:

> [. . .] Achei que está bem claro e cobrindo todos os testes desta área. E2

> [. . .] A decomposição está bem clara e eu não adicionaria e nem removeria nenhuma destas classificações. E6

6.3.3 Teste de estresse

Em relação a classificação de **teste de estresse**, alguns especialistas acharam que a decomposição desta classificação ficou pequena, mas que ao mesmo tempo também concordaram entre si que este tipo de teste é mais complicado para se classificar. Outros especialistas informaram não ter conhecimento sobre este tipo de teste. O comentário a seguir retrata bem a análise desta classificação:

> [. . .] Eu achei que ficou um pouco pequena esta decomposição, talvez pudesse cobrir mais áreas dos testes de estresse, como por exemplo os testes com intuito de encontrar ***crashes*** **(quebra do sistema)** a partir do excesso de processamento do sistema. E este tipo de teste não me parece entrar em nenhumas destas categorias. E3

6.3.4 Teste de desempenho

No que se refere a classificação de **teste de desempenho,** alguns especialistas acharam sua decomposição bem clara e que não adicionariam ou removeriam nenhuma classificação. Entretanto, tiveram alguns especialistas que acharam ambíguo a classificação de **tempo de resposta** com a de **medição**. As transcrições a seguir descrevem esta análise:

> [. . .] Eu achei tempo de resposta e medição muito parecidos. Acredito que poderia fazer uma junção dos dois. E2

> [. . .] Eu achei bem claro a decomposição e não adicionaria ou removeria nenhuma classificação. E6

6.3.5 Teste de interface gráfica do usuário

Com relação a classificação de **teste de interface gráfica do usuário**, um especialista relatou que trabalha com testes voltados para **localização** do software, que é o processo de traduzir um texto garantindo que ele terá todos os componentes para atender as necessidades culturais e linguísticas de um país específico, e relatou que das classificações propostas de teste de interface gráfica do usuário, nenhuma delas se encaixaria neste tipo de teste. A transcrição do áudio deste especialista retrata isso:

> [. . .] Testes de **globalização** ou **internacionalização** são os testes para garantir que o software está devidamente localizado em todos os idiomas suportados. A princípio eu achei que teste de globalização iria entrar neste nível da taxonomia, porém nada dessa decomposição cobre essa parte de globalização. E4

6.3.6 Teste de aceitação

Se tratando da classificação de **teste de aceitação**, todos concordaram que a sua decomposição está bem definida, e que não adicionariam ou removeriam nenhuma classificação. Os comentários a seguir descrevem esta análise:

> [. . .] Está bem fechado, não adicionaria ou removeria nenhuma das duas classificações. E1

> [. . .] Ficou bem claro esta decomposição e está cobrindo todas as áreas dos testes de aceitação. E2

6.3.7 Teste beta

Com relação a classificação de **teste beta**, a análise é a mesma da anterior, todos os especialistas concordaram que a sua decomposição ficou clara, e que não removeriam ou adicionariam nenhuma classificação. Assim como na anterior, o debate para esta decomposição não durou muito, e a transcrição do áudio a seguir retrata isto:

> [. . .] Está bem clara a decomposição e não adicionaria ou removeria nenhumas das duas classificações. E5

6.3.8 Teste exploratório

No que se refere a classificação de **teste exploratório**, os especialistas inicialmente concordaram que esta classificação estava bem completa e cobrindo tudo,

inclusive um deles informou que tinha mais testes do que conhecia. Entretanto, um dos especialistas levantou o debate sobre a classificação de **teste de fumaça por intuição e experiência**, pois este tipo de teste poderia ser encaixado na primeira classificação de teste de funcional e de sistema, que é o **teste básico**, ao invés de estar na categoria de teste exploratório. Por fim, todos concordaram nesta mudança e os comentários a seguir relatam esta análise:

> [. . .] Está bem completo e com mais testes do que eu conhecia da parte de exploratório. E1

> [. . .] Teste de fumaça por intuição e experiência já está coberto pela parte de teste básico na categoria de teste funcional e de sistema e não se encaixa em teste exploratório. E6

6.3.9 Teste de fronteira

Se tratando da classificação de **teste de fronteira**, ocorreram dúvidas com relação as duas seguintes classificações: **pior caso** e **pior caso robusto**, os especialistas acharam isto ambíguo e que poderiam ser colocados em apenas uma classificação. As transcrições dos áudios a seguir descrevem isto:

> [. . .] Pior caso e pior caso robusto acredito que poderia ser uma classificação só. E2

> [. . .] Pior caso já cobre, como o nome diz, o pior caso, então não teria sentido adicionar um pior caso robusto, pois nesse caso, o pior caso robusto já seria o pior caso. E3

6.4 Limitações e ameaças à validade

Com relação as limitações do grupo focal, pode se destacar o fato de que a taxonomia engloba muitos tipos de teste de caixa preta, houve participantes que não tinham conhecimento sobre alguns tipos de teste. Entretanto, por se tratar de um grupo de 6 participantes, teve o fato de que quando um ou mais não entendiam sobre determinado tipo de teste, outro(s) participante(s) tinha(m) conhecimento.

Outra limitação está associada ao tempo limitado para esta atividade, o que poderia ocorrer de os participantes não conseguirem entender alguma parte da composição da taxonomia. Para tentar evitar isso, o moderador realizou toda uma explicação prévia sobre o que é e qual objetivo do grupo focal e da ideia central da proposta.

Se tratando de ameaça à validade, teve o fato de que o autor desta proposta foi o mediador do grupo focal, e com isso, os participantes poderiam acabar sendo mais cautelosos em suas opiniões. Para evitar isto, buscou-se selecionar um maior número de participantes que não tinham relação direta com o autor.

6.5 Considerações finais

O grupo focal teve como objetivo coletar o máximo possível de feedback sobre a proposta desta pesquisa, e com isso, poder validar se a pesquisa tem relevância e é válida. A técnica foi realizada com um grupo de participantes de naturezas parecidas em se tratando do tema da pesquisa.

A partir do objetivo do grupo focal, que foi de avaliar a taxonomia de casos de teste de software de caixa preta sob os aspectos de **completude**, **clareza** e **adequação** da estrutura proposta, foram consolidadas as seguintes sugestões com base nas ideias levantadas pelos participantes:

1) Detalhar e destrinchar mais a decomposição da classificação sobre teste de estresse;
2) Fazer a junção da classificação de teste de resposta e de medição;
3) Incluir uma classificação para testes de globalização/internacionalização dentro da categoria de testes de interface gráfica do usuário ou em uma outra categoria;
4) Remover teste de fumaça por intuição e experiência da categoria de exploratório;
5) Fazer a junção da classificação de pior caso e de pior caso robusto.

Com base nos relatos apresentados, foi possível constatar a relevância da pesquisa e que há indícios de sua validade.

7 CONCLUSÃO

Este capítulo tem como objetivo apresentar as considerações finais e contribuições resultantes desta pesquisa, assim como os possíveis trabalhos futuros.

7.1 Considerações finais e contribuições

Este trabalho teve como objetivo desenvolver uma classificação de casos de teste de software de caixa preta com o intuito de auxiliar os engenheiros de testes em suas execuções. Foi realizado uma pesquisa exploratória sobre temas como qualidade de software, teste de software, classificação e taxonomia, com o intuito de se obter um maior conhecimento sobre o tema da pesquisa e auxiliar o autor no decorrer do trabalho.

Por meio dos trabalhos relacionados a esta pesquisa, é possível notar que é um tema relativamente recente, tendo o autor encontrado a pesquisa mais antiga datada de 2008. Os trabalhos mencionados tinham como proposta uma classificação pequena de diferentes tipos de testes. Neste trabalho, a taxonomia desenvolvida abrange um número muito maior de testes se comparado com os trabalhos relacionados (tendo como foco principal, os teste de caixa preta).

A principal contribuição deste trabalho foi uma taxonomia de casos de teste de software de caixa preta. Esta taxonomia é estruturada em 2 níveis (primeiro nível e nível subsequente) onde foi classificado os principais casos de teste de software de caixa preta e teve como base as propostas de vários autores da área. Foi realizado um grupo focal com o intuito de coletar dados qualitativos sobre a proposta, e por meio desta técnica, foi possível coletar dados importantes sobre a validade e relevância desta pesquisa, assim como sugestões de possíveis melhorias.

7.2 Trabalhos futuros

Como trabalho futuro, pode-se destacar a expansão dessa taxonomia, incluindo mais itens em cada nível ou até mesmo expandindo o número de níveis da mesma. Lembrando que a estrutura da taxonomia proposta não tem a intenção de esgotar as possibilidades apresentadas entre os diversos tipos de teste de caixa preta, mas servir como um direcionamento para o pensamento sistematizado a respeito do domínio aqui apresentado.

Vale também destacar como trabalho futuro, além da ampliação da taxonomia, realizar também uma validação de maior escalar. Como por exemplo, realizar um estudo

de caso em uma empresa de tecnologia, utilizando a taxonomia em um ambiente real. Desta forma, mais dados poderiam ser coletados com relação a taxonomia desenvolvida, e assim, melhor cada vez mais a mesma.

Um outro trabalho futuro é poder ampliar a taxonomia para abordar não só testes de caixa preta, mas também outros tipos de teste como os testes de caixa branca, testes de caixa cinza, testes não funcionais, entre outros.

Referências

ANANTHARAM, V. The Optimal Buffer Allocation Problem. *IEEE TRANSACTIONS ON INFORMATION THEORY*, v. 35, n. 4, p. 721 – 725, Julho 1989.

ANSI STANDARD (ANSI/ASQC). *A Quality Management Standard for Environmental Programs*. [S.l.], 1994.

ASSOCIACÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE. Mercado Brasileiro de Software - Panorama e Tendências. 2017.

BABBIE, E. R. *The basics of social research*. [S.l.]: Wadsworth Cengage, 2014.

BACH, J. Exploratory Testing Explained. v. 1.3, Abril 2003.

BARESI, L.; PEZZÈ, M. An Introduction to Software Testing. *Electronic Notes in Theoretical Computer Science*, v. 148, p. 89 – 111, Fevereiro 2006.

BARTIÉ, A. *Garantia Da Qualidade De Software*: Adquirindo Maturidade Organizacional. [S.l.]: CAMPUS, 2013.

BASTOS, A. et al. *Base de conhecimento em teste de software*. [S.l.]: Martins Fontes, 2007.

BAYONA-ORÉ, S. et al. Critical success factors taxonomy for software process deployment. *Software Quality Journal*, Hingham, v. 22, p. 21 – 48, Março 2014.

BERG, B. L.; LUNE, H. *Qualitative research methods for the social sciences*. [S.l.]: Boston Pearson, 2012.

BERTOLINO, A. Software Testing Research: Achievements, Challenges, Dreams. *Future of Software Engineering (FOSE)*, Washington, p. 85 – 103, Maio 2007.

CAPLAN, S. Using focus group methodology for ergonomic design. *Ergonomics*, v. 33, p. 527 – 533, 1990.

CENTRAL STATISTICS OFFICE. *What is a Classification?* 2018. Disponível em: <https://www.cso.ie/en/methods/classifications/whatisaclassification/>. Acesso em: 29/07/2018.

CHANDRUPATLA, T. R. Quality Concepts. In: ____. *Quality and Reliability in Engineering*. [S.l.: s.n.], 2009. cap. 1.

CMMI/SEI. *Capability Maturity Model Integration for Developtment*. v1.3. [S.l.], 2010.

CROSBY, P. B. *Quality is free*: the art of making quality certain. [S.l.]: New York: McGraw-Hill, 1979.

DEMING, W. E. *Out of the crisis*. Cambridge: MIT Center for Advanced Engineering, 1982.

DENZIN, N. K.; LINCOLN, Y. S. *The SAGE Handbook of Qualitative Research*. 3. ed. [S.l.]: SAGE Publications, 2005.

ENGSTRÖM, E.; RUNESON, P.; SKOGLUND, M. A systematic review on regression test selection techniques. *Information and Software Technology*, Newton, p. 14 – 30, Janeiro 2010.

FEIGENBAUM, A. V. *Total Quality Control*. [S.l.]: McGraw-Hill, 1991.

FELDERER, M.; AGREITER, B.; ZECH, P. A Classification for Model-Based Security Testing. In: *The Third International Conference on Advances in System Testing and Validation Lifecycle*. [S.l.: s.n.], 2011.

FELDERER, M.; FOURNERET, E. A systematic classification of security regression testing approaches. *International Journal on Software Tools for Technology Transfer (STTT)*, Berlin, p. 305 – 319, Junho 2015.

FELDERER, M. et al. Model-based security testing: a taxonomy and systematic classification. *Software Testing, Verification & Reliability*, Chichester, p. 119 – 148, Março 2016.

FREITAS, A. H. REFLEXÕES SOBRE A PESQUISA ACADÊMICA: REVISÃO BIBLIOGRÁFICA, VIVÊNCIA E CONHECIMENTO. *Palíndromo*, v. 8, n. 15, p. 74 – 82, Junho 2016.

FURTADO, A. P. C. C. *FAST*: UM FRAMEWORK PARA AUTOMAÇÃO DE TESTE. 2017. 178 p. Tese (Doutorado em Ciência da Computação) — Universidade Federal de Pernambuco, Recife.

GERMAN INDUSTRY STANDARD DIN 55350. *Concepts for quality management and statistics*. [S.l.], 2004.

GIVEN, L. M. *The Sage Encyclopedia of Qualitative Research Methods*. [S.l.]: SAGE Publications, 2008.

IEEE STANDARD. *IEEE Standard for Software Quality Assurance Processes*. [S.l.], 2014.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. *IEEE Standard Computer Dictionary - A Compilation of IEEE Standard Computer Glossaries*. [S.l.], 1990.

ISO/IEC/IEEE 24765. *Systems and software engineering - Vocabulary*. [S.l.], 2010.

ISO/IEE/829. *IEEE Standard for Software and System Test Documentation*. Nova Iorque, 2008.

ITKONEN, J. Exploratory and Experience Based Testing. *Aalto University School of Science*, Novembro 2011.

ITKONEN, J.; MÄNTYLÄ, M. V.; LASSENIUS, C. How Do Testers Do It? An Exploratory Study on Manual Testing Practices. *Third International Symposium on Empirical Software Engineering and Measurement*, p. 494 – 497, Outubro 2009.

JORGENSEN, P. C. *Software Testing A Craftsman's Approach*. 4. ed. [S.l.]: CRC Press, 2014.

JURAN, J. M. How to Think about Quality. In: ____. *JURAN'S QUALITY HANDBOOK*. [S.l.]: McGraw-Hill, 1998. cap. 2.

KAIKKONEN, A. et al. Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing. *JOURNAL OF USABILITY STUDIES*, v. 1, p. 4 – 16, Novembro 2005.

KASURINEN, J.; TAIPALE, O.; SMOLANDER, K. Test case selection and prioritization: risk-based or design-based? In: *International Symposium on Empirical Software Engineering and Measurement (ESEM)*. Italy: [s.n.], 2010.

KITZINGER, J. The methodology of Focus Groups: the importance of interaction between research participants. *Sociology of Health & Illnes*, v. 16, n. 1, p. 103 – 120, Janeiro 1994.

KONTIO, J. Using the focus group method in software engineering: obtaining practitioner and user experiences. In: *International Symposium on Empirical Software Engineering*. Redondo Beach: [s.n.], 2004.

KOSCIANSKI, A.; SOARES, M. dos S. Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. In: ____. *Qualidade de Software*. [S.l.]: Novatec, 2007.

LANGRIDGE, D. *Classificação*: abordagem para estudantes de biblioteconomia. Rio de Janeiro: Interciência, 1977.

LAWANNA, A. An Effective Test Case Selection for Software Testing Improvement. In: *Computer Science and Engineering Conference (ICSEC), 2015 International*. Chiang Mai: [s.n.], 2015.

LEAL, A. L. de C. *UMA PROPOSTA DE TAXONOMIA DE BOAS PRÁTICAS EM DESENVOLVIMENTO DE SOFTWARE*. 2009. 99 p. Dissertação (Ciência da Computação) — Universidade Federal de Viçosa.

LINNÉ, K. V. *Systema Naturae*. [S.l.]: LAURENTII SALVII, 1735.

LUFT, C. C. *TESTE DE SOFTWARE: UMA NECESSIDADE DAS EMPRESAS*. 2012. 91 p. Monografia (Bacharelado em Ciência da Computação) — UNIJUÍ – UNIVERSIDADE REGIONAL DO NOROESTE DO ESTADO DO RIO GRANDE DO SUL.

MPS.BR. *Guia de Implementação – Parte 4*: Fundamentação para Implementação do Nível D do MR-MPS-SW:2016. [S.l.], 2016.

MYERS, G. J. *The Art of Software Testing*. [S.l.]: TOM BADGETT, 1979.

NAIK, K.; TRIPATHY, P. *SOFTWARE TESTING AND QUALITY ASSURANCE*. [S.l.]: Wiley, 2008.

NATIONAL INFORMATION STANDARDS ORGANIZATION (NISO). *Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*. Baltimore, 2005.

NIDHRA, S.; DONDETI, J. BLACK BOX AND WHITE BOX TESTING TECHNIQUES - A LITERATURE REVIEW. *International Journal of Embedded Systems and Applications (IJESA)*, v. 2, n. 2, Junho 2012.

NOVO, H. F. A TAXONOMIA ENQUANTO ESTRUTURA CLASSIFICATÓRIA: UMA APLICAÇÃO EM DOMÍNIO DE CONHECIMENTO INTERDISCIPLINAR. 2010.

NUNES, L. *DA CLASSIFICAÇÃO DAS CIÊNCIAS À CLASSIFICAÇÃO DA INFORMAÇÃO uma análise do acesso ao conhecimento*. 2007. 124 p. Dissertação (Ciência da Informação) — Pontifícia Universidade Católica de Campinas, Campinas.

PIEDADE, M. A. R. *Introdução à teoria da classificação*. [S.l.]: Interciência, 1983.

PRESSMAN, R. S. *Engenharia de Software Uma Abordagem Profissional*. [S.l.]: McGraw-Hill, 2011.

SANTOS, R. E. et al. Building a theory of job rotation in software engineering from an instrumental case study. In: ACM, 2016. *Proceedings of the 38th International Conference on Software Engineering*. [S.l.], 2016. p. 971 – 981.

SIQUEIRA, H. da C. et al. MODELAGEM DE PROCESSO E TAXONOMIA: ferramentas para a organização da informação e do conhecimento organizacional. In: *XVII - Encontro Regional dos Estudantes de Biblioteconomia, Documentação, Ciência e Gestão da Informação - EREBD*. Fortaleza: [s.n.], 2014.

SOMMERVILLE, I. *Engenharia de Software*. [S.l.]: Pearson, 2011.

STEBBINS, R. A. *Exploratory Research in the Social Sciences*. [S.l.]: Sage Publications, 2001.

TERRA, J. C. C. et al. Taxonomia: Elemento Fundamental para a Gestão do Conhecimento,. 2004.

TUTEJA, M.; DUBEY, G. A Research Study on importance of Testing and Quality Assurance in Software Development Life Cycle (SDLC) Models. *International Journal of Soft Computing and Engineering (IJSCE)*, v. 2, Julho 2012.

UTTING, M.; PRETSCHNER, A.; LEGEARD, B. A taxonomy of model-based testing approaches. *Software Testing, Verification & Reliability*, Chichester, p. 297 – 312, Agosto 2012.

VAUGHN, S.; SCHUMM, J. S.; SINAGUB, J. *Focus Group Interviews in Education and Psychology*. [S.l.: s.n.], 1996.

VILLALÓN, J. C. et al. A taxonomy for software testing projects. In: *Information Systems and Technologies (CISTI)*. Aveiro: [s.n.], 2015.

WILKINS, J. S. *What is systematics and what is taxonomy?* 2011. Disponível em: <http://evolvingthoughts.net/2011/02/05/what-is-systematics-and-what-is-taxonomy/>. Acesso em: 04/02/2018.

Apêndices

APÊNDICE A – Taxonomia de Casos de Teste de Software de Caixa Preta (Visão Completa)