

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO NORTE

LISIANE GONÇALVES DA SILVA

**ANÁLISE DAS ATIVIDADES DE TESTE DE SOFTWARE QUANTO A SUA
APLICABILIDADE E IMPORTÂNCIA EM EMPRESAS PRIVADAS**

Nova Cruz/RN

2022

LISIANE GONÇALVES DA SILVA

**ANÁLISE DAS ATIVIDADES DE TESTE DE SOFTWARE QUANTO A SUA
APLICABILIDADE E IMPORTÂNCIA EM EMPRESAS PRIVADAS**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Me. Aislânia Alves de Araujo

Nova Cruz/RN

2022

Catálogo da publicação na fonte
Biblioteca do Instituto Federal de Educação, Ciência e Tecnologia do RN
Campus Nova Cruz

S586a SILVA, Lisiane Gonçalves da.

Análise das atividades de teste de software quanto a sua aplicabilidade e importância em empresas privadas. / Lisiane Gonçalves da Silva. – Nova Cruz/RN, 2022.
53f.

Orientador: Me. Aislânia Alves de Araujo - Monografia (Monografia em ciências exatas e da terra). – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – Nova Cruz/RN, 2022.

1. Engenharia de software – Monografia. 2. Teste de software – Monografia. 3. Programação - Monografia. I. ARAUJO, Aislânia Alves de. II. Título.

IFRN

CDU: 004.41(0813.2)

LISIANE GONÇALVES DA SILVA

**ANÁLISE DAS ATIVIDADES DE TESTE DE SOFTWARE QUANTO A SUA
APLICABILIDADE E IMPORTÂNCIA EM EMPRESAS PRIVADAS**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Trabalho aprovado. Nova Cruz/RN, 29 de Março de 2022.

BANCA EXAMINADORA

Me. Aislânia Alves de Araujo
Orientadora - IFRN

Me. Lucas Vieira de Souza
Examinador - IFRN

Me. Maria Jane de Queiroz
Examinadora - IFRN

AGRADECIMENTOS

Agradeço primeiramente a Deus que me deu forças para superar cada obstáculo, sem ele seria impossível chegar aqui. Aos meus pais, Maria Auxiliadora e Eleomar, a minhas irmãs, Lidiane e Maria Dasdores, que me deram total apoio durante todo o curso.

À minha amiga, e também parceira de turma, Milena da Silva Alves, que sempre esteve comigo. Aos meus amigos Gileno cordeiro, Isabele Oliveira e Walquiria Gomes por toda parceria durante esse percurso.

À minha amiga Elizabete Jeronimo, também por todo apoio e incentivo.

Ao Instituto Federal do Rio Grande do Norte, por disponibilizar uma estrutura e um ensino de qualidade.

À minha orientadora, Aislânia Alves de Araújo, por todo comprometimento, paciência e disponibilidade.

À toda equipe docente do curso, pelo apoio e conhecimento repassado, contribuindo significativamente com a minha formação.

RESUMO

As atividades de testes são fundamentais no processo de desenvolvimento de software, elas buscam garantir a qualidade do produto final, sendo a qualidade de software um aspecto predominante no mercado de desenvolvimento. Deste modo, o presente trabalho objetiva analisar as atividades de testes de software, quanto a sua aplicabilidade e importância, em empresas do âmbito privado. Trata-se de uma pesquisa descritiva de método quantitativo, realizada com a participação de cinco empresas desenvolvedoras de software do Rio Grande Norte, especificamente nas cidades de Caicó, Natal e Nova Cruz. O levantamento de dados ocorreu por meio de um questionário adaptado com o total de vinte e duas questões. A partir da análise dos dados coletados foi constatado que apesar das empresas envolvidas realizarem o testes de software, elas não realizam as atividades voltadas para a documentação do processo, o levantamento de riscos e a conservação dos testes realizados, o que pode comprometer a qualidade do software.

Palavras-chave: Empresas privadas; Engenharia de software; Qualidade de software; Teste de software.

ABSTRACT

Testing activities are fundamental in the software development process, they seek to guarantee the quality of the final product, with software quality being a predominant aspect in the development market. Thus, the present work aims to analyze software testing activities, regarding their applicability and importance in private companies. This is a descriptive research using a quantitative method, carried out with the participation of five software development companies in Rio Grande Norte, specifically in the cities of Caicó, Natal and Nova Cruz. Data collection took place through an adapted questionnaire with a total of twenty-two questions. From the analysis of the collected data, it was found that although the companies involved carry out the software tests, they do not carry out the activities aimed at documenting the process, surveying risks and preserving the tests performed, which can compromise the quality of the software.

Keywords: Private companies; Software engineering; Software quality; Software testing.

LISTA DE FIGURAS

Figura 1 – Camadas da Engenharia de Software	15
Figura 2 – Modelo Cascata.	17
Figura 3 – Modelo V	18
Figura 4 – Modelo Incremental	19
Figura 5 – Engenharia de Software Baseada em Reuso	19
Figura 6 – Média dos Dados Quanto a Aplicabilidade	35
Figura 7 – Média dos Dados Quanto a Importância	42
Figura 8 – Correlação Entre as Média Obtidas: Aplicabilidade e Importância	43

LISTA DE TABELAS

Tabela 1 – Descrição da Escala de Aplicabilidade e Importância do Questionário Aplicado	29
Tabela 2 – Percentual das Respostas Obtidas Para Cada Variável, Baseado na Aplicabilidade	30
Tabela 2 – Continuação	31
Tabela 2 – Continuação	32
Tabela 3 – Análise Estatística dos Dados Referente a Aplicabilidade	33
Tabela 3 – Continuação	34
Tabela 4 – Percentual das Respostas Obtidas Para Cada Variável, Baseado na Importância	36
Tabela 4 – Continuação	37
Tabela 4 – Continuação	38
Tabela 4 – Continuação	39
Tabela 5 – Análise Estatística dos Dados Referente a Importância	39
Tabela 5 – Continuação	40
Tabela 5 – Continuação	41
Tabela 6 – Relação Entre os Resultados Obtidos: Aplicabilidade e Importância . .	43
Tabela 6 – Continuação	44
Tabela 6 – Continuação	45

SUMÁRIO

1	INTRODUÇÃO	11
1.1	CONTEXTUALIZAÇÃO	11
1.2	OBJETIVOS	12
1.2.1	Objetivo Geral	12
1.2.2	Objetivos Específicos	12
1.3	JUSTIFICATIVA	12
1.4	ESTRUTURA DO TRABALHO	12
2	REFERENCIAL TEÓRICO	14
2.1	ENGENHARIA DE SOFTWARE	14
2.1.1	Processo de Desenvolvimento de Software	15
2.1.1.1	Modelo Cascata	16
2.1.1.2	Desenvolvimento Incremental	18
2.1.1.3	Integração e Configuração	19
2.2	QUALIDADE DE SOFTWARE	20
2.2.1	Processo de Garantia da Qualidade de Software	20
2.3	TESTE DE SOFTWARE	22
2.3.1	Técnicas de Teste	23
2.3.1.1	Teste Funcional	23
2.3.1.2	Teste Estrutural	24
2.3.2	Níveis de teste	24
2.3.2.1	Teste de Unidade	25
2.3.2.2	Teste de Integração	25
2.3.2.3	Teste de Sistema	25
2.3.2.4	Teste de Aceitação	25
2.3.3	Processo de Teste de Software	26
2.3.3.1	Planejamento de Teste	26
2.3.3.2	Casos de Teste	26
2.3.3.3	Execução do Teste	26
2.3.3.4	Avaliação dos Resultados de Teste	27
3	PROCEDIMENTOS METODOLÓGICOS	28
3.1	CARACTERIZAÇÃO DO OBJETO DE ESTUDO	28
3.2	MÉTODO DA PESQUISA	28
3.3	APRESENTAÇÃO E ANÁLISE DE DADOS	29

3.3.1	Análise Baseada na Aplicabilidade	29
3.3.2	Análise Baseada na Importância	36
3.3.3	Correlação Entre os Resultados Obtidos	42
4	CONSIDERAÇÕES FINAIS	47

REFERÊNCIAS

APÊNDICES	50
APÊNDICE A QUESTIONÁRIO APLICADO	51

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

De acordo com Noello (2016), a qualidade determina a competitividade de mercado no desenvolvimento de software. Dessa forma é importante que as empresas desenvolvedoras de software busquem estabelecer um processo que venha a alcançar um produto de qualidade.

A qualidade de software refere-se ao fato de que o produto precisa atender as expectativas do cliente a partir dos requisitos determinados no início do projeto (PERUCCI; CAMPOS, 2016).

Segundo Kasse (2008 apud SCATOLINO; CAMILO, 2019), na engenharia de software a qualidade do processo de desenvolvimento influencia significativamente na qualidade do software. Este processo possui um comportamento flexível, o que permite à equipe buscar e estabelecer práticas adequadas para o desenvolvimento do produto (BRAGA, 2019).

O processo de desenvolvimento de software requer muita atenção às atividades realizadas e por isso a engenharia de software busca acompanhar tal processo para a obtenção de um produto de qualidade. Para garantir a qualidade de software a aplicação dos testes é um dos fatores primordiais, por meio dos quais é possível aumentar a chance de sucesso do produto desenvolvido, correspondendo às expectativas de seus desenvolvedores e usuários.

Apesar da importância das atividades de testes softwares, não são todas as empresas de desenvolvimento que as põem em prática, geralmente por questões de tempo, custo, ferramentas adequadas e falta de profissionais qualificados. Estas atividades quando não realizadas, ou realizadas de forma inadequada, podem comprometer a qualidade do produto (SANTOS, 2016).

Partindo dessa problemática, o presente trabalho pretende avaliar a aplicação do teste de software e qual a sua relevância na busca de um produto de qualidade.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Analisar a aplicabilidade e a importância das atividades de testes de software no processo de desenvolvimento em empresas do âmbito privado.

1.2.2 Objetivos Específicos

- Compreender a importância dos testes software para a qualidade do software;
- Analisar a realização de testes de software por empresas privadas de desenvolvimento de software;
- Verificar se existem e como estão organizadas a documentação, o levantamento de riscos e a conservação dos testes de software por empresas privadas do ramo de desenvolvimento de software.

1.3 JUSTIFICATIVA

Tendo em vista que o teste de software é uma das atividades da engenharia de software que visa a qualidade do produto final, é importante que as empresas desenvolvedoras o incluam no processo de desenvolvimento. Com isso, este trabalho busca, a partir de seu objetivo, expor como as empresas lidam com as atividades de testes atualmente, assim contribuindo para o enriquecimento da literatura a respeito da área abordada.

Para alcançar o objetivo do trabalho, optou-se pela aplicação de um questionário, o qual analisa as atividades de testes de acordo com seu grau de aplicabilidade e importância no processo de desenvolvimento de software realizado pelas empresas participantes.

1.4 ESTRUTURA DO TRABALHO

O trabalho está organizado da seguinte forma: o capítulo 1, composto pela introdução, abordando os objetivos, a justificativa e o método da pesquisa. O capítulo 2 se trata da revisão bibliográfica para a fundamentação do trabalho, a qual aborda os conceitos de engenharia, qualidade e teste de software. O capítulo 3 refere-se aos procedimentos

metodológicos da pesquisa, tais como a caracterização do objeto de estudo, método, a apresentação e análise dos resultados. Por fim, o capítulo 4 traz as considerações finais.

2 REFERENCIAL TEÓRICO

2.1 ENGENHARIA DE SOFTWARE

Com o passar dos anos o Software vem evoluindo e ganhando maior importância, o que era apenas um produto computacional de análise e resolução de problemas, passou a atuar como sistema operacional e de comunicação (PRESSMAN; MAXIM, 2016), dos quais, atualmente, as empresas e/ou organizações, por exemplo, precisam para a automatização de suas atividades, sejam elas de pequeno ou grande porte, públicas ou privadas. A partir dessa evolução, o processo de desenvolvimento de software tornou-se mais complexo (VALENTE, 2020). Dessa forma, a Engenharia de Software propõe solucionar os problemas identificados durante tal processo.

A engenharia de software se trata da área da computação que se preocupa com os fatores do desenvolvimento de software, desde o início de sua construção até a sua fase de execução e manutenção (SOMMERVILLE, 2018), contribuindo assim com a qualidade do produto final.

Segundo Pressman e Maxim (2016), a engenharia de software é rica em técnicas, métodos e ferramentas capazes de melhorar o processo de desenvolvimento do software e até mesmo o produto resultante, e sua importância está na capacidade de desenvolver sistemas complexos de qualidade dentro do prazo estimado, em impor disciplina no processo e permitir que o desenvolvedor realize o trabalho de acordo com as suas necessidades. Ele ainda afirma que esta “se trata de uma tecnologia em camadas”, a qual deve manter seu foco na qualidade, como apresentada na Figura 1.

Como visto na Figura 1, Pressman e Maxim (2016) definem a engenharia de software em 4 camadas:

Ferramentas: Responsável por fornecer suporte automatizado ou semi automatizado aos métodos e processos.

Métodos: Responsável pelo fornecimento de informações técnicas para o desenvolvimento do software. Modelagem de projeto, comunicação, análise dos requisitos e testes são algumas de suas atividades.

Processo: base da engenharia de software responsável por manter as camadas de tecnologia coesas e permitir que o software seja desenvolvido dentro do prazo e de forma racional.

Figura 1 – Camadas da Engenharia de Software



Fonte: Adaptado de Pressman e Maxim (2016, p. 16)

Foco na qualidade: camada que sustenta a engenharia de software.

Na aplicação da engenharia de software estão envolvidos os processos de softwares, por meio dos quais os desenvolvedores são capazes de avaliar e aplicar estratégias que possam contribuir com a qualidade do sistema. Além disso, é necessário que mecanismos de gerência e planejamento sejam ofertados ao processo de desenvolvimento (FERNANDES, 2017). Tais processos serão abordados no próximo tópico.

2.1.1 Processo de Desenvolvimento de Software

O processo de desenvolvimento de software refere-se a um conjunto de atividades relacionadas e indispensáveis no desenvolvimento de sistemas. Esse processo se dá diante as necessidades do cliente e a aptidão do desenvolvedor, porém é necessário que todo processo inclua as atividades principais da engenharia de software: especificação, desenvolvimento, validação e evolução (SOMMERVILLE, 2018).

Apesar destas atividades indicarem, conseqüentemente, uma ordem de prioridade entre elas, sua organização pode ocorrer de diferentes formas, sem que uma etapa precise ser finalizada para dar início a próxima (FERNANDES, 2017). Essa organização vai variar de acordo com o processo de desenvolvimento definido:

É importante ter em mente que o processo de software adotado muda de acordo com o projeto a ser desenvolvido. Isso significa que cada projeto

exigirá ajustes específicos e direcionados à aplicação a ser desenvolvida. A cada novo projeto, os cenários mudam: tecnologias, pessoas, rotinas, processos, procedimentos, atividades. Logo, mudam também os processos de software adotados no desenvolvimento (ALÉSSIO; SABADIN; ZANCHETT, 2017).

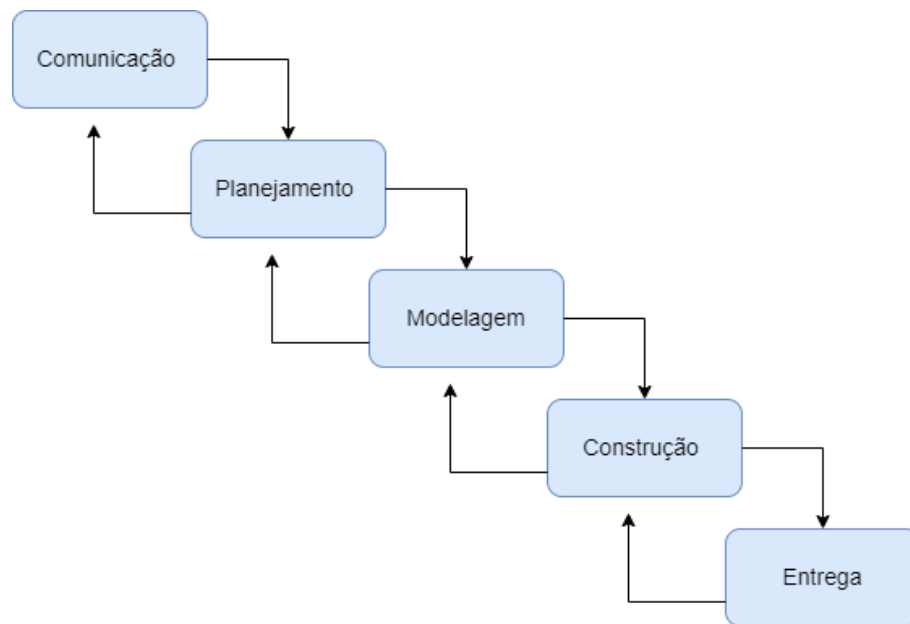
Os modelos de processo do ciclo de vida do software são definidos com o intuito de representar algumas estruturas do processo de desenvolvimento de software (FERNANDES, 2017).

De acordo com Sommerville (2018), são abordados três processos de softwares genéricos, os quais são possíveis de adaptar ou ampliar, gerando processos de engenharia de software mais específicos, são eles: modelo cascata, desenvolvimento incremental, integração e configuração, responsáveis respectivamente por representar as atividades essenciais do processo, intercalar as atividades de especificação, desenvolvimento e validação, e pela disponibilidade de componentes ou sistemas reusáveis.

2.1.1.1 Modelo Cascata

Como citado anteriormente o modelo cascata é o responsável pelas principais atividades do processo de software. Pressman e Maxim (2016) a define como uma abordagem sistemática e sequencial no desenvolvimento de sistemas que parte desde o levantamento de requisitos, por meio do cliente, até a sua fase de conclusão, como apresentado na Figura 2 a seguir.

Figura 2 – Modelo Cascata.

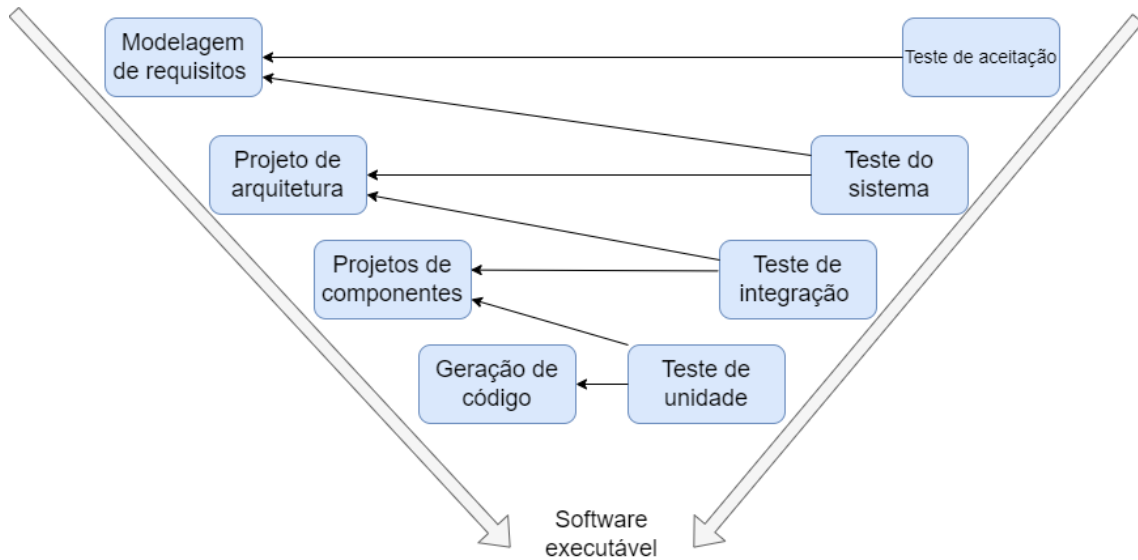


Fonte: Adaptado de Pressman e Maxim (2016, p. 42)

Este é um dos modelos de processos conduzidos por plano, sendo necessário que haja um planejamento e a criação de um cronograma de atividades antes de dar início ao processo de desenvolvimento do software (SOMMERVILLE, 2018).

Uma das variações do modelo em cascata é o modelo V, apresentado na Figura 3, o qual destaca a relação entre as atividades do processo e as atividades de teste. Uma característica destes modelos é a entrega do produto em sua totalidade, depois da realização de todas as atividades esperadas (ALÉSSIO; SABADIN; ZANCHETT, 2017).

Figura 3 – Modelo V

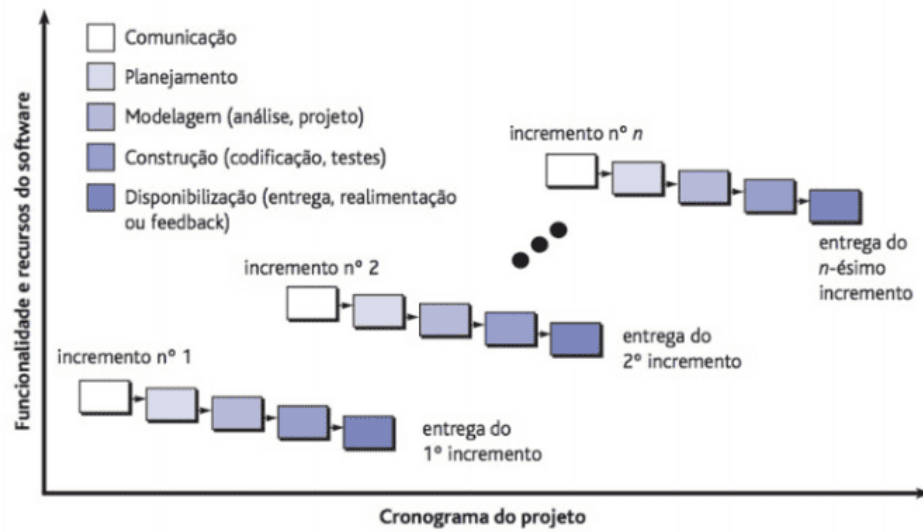


Fonte: Adaptado de Pressman e Maxim (2016, p. 43)

2.1.1.2 Desenvolvimento Incremental

Em aplicações maiores, é possível que o modelo de processo sequencial não atenda às suas necessidades, mesmo diante de um levantamento de requisitos adequado. Por este motivo é viável a utilização do modelo incremental, o qual permite que o sistema seja separado por módulos ou subsistemas (ALÉSSIO; SABADIN; ZANCHETT, 2017). Sua ideia está na produção e entrega de uma versão operacional do sistema para que o cliente possa avaliá-lo a cada ciclo de interação durante seu desenvolvimento, conforme mostra a Figura 4, até que o objetivo do sistema seja alcançado (FERNANDES, 2017).

Figura 4 – Modelo Incremental

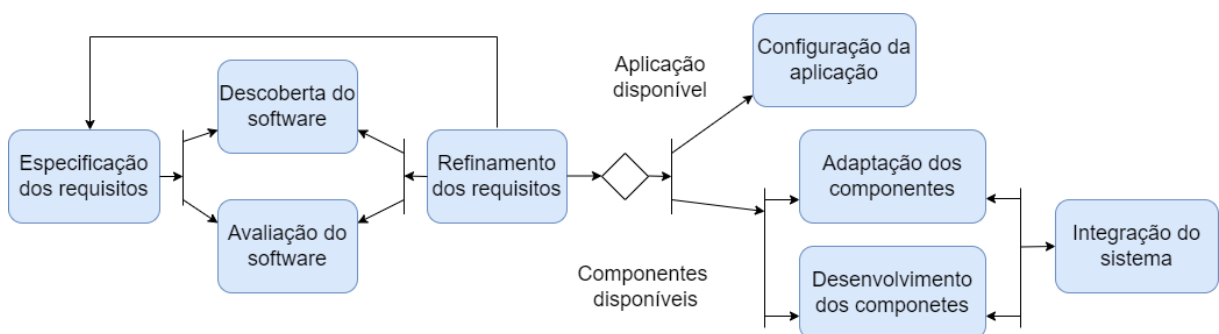


Fonte: Adaptado de Pressman e Maxim (2016, p. 44)

2.1.1.3 Integração e Configuração

Na maior parte dos projetos existe a reutilização de software, os quais são modificados a partir da necessidade do novo projeto, esse processo de reutilização se dá geralmente em torno de interação e configuração seguindo as seguintes etapas: especificação de requisitos, descoberta e avaliação do software, refinamento dos requisitos, configuração da aplicação, a adaptação e integração dos componentes, como ilustra a Figura 5. Assim reduzindo a quantidade de software a ser desenvolvido, riscos e custos, além da entrega mais rápida do produto (SOMMERVILLE, 2018).

Figura 5 – Engenharia de Software Baseada em Reuso



Fonte: Adaptado de Sommerville (2018, p. 38)

2.2 QUALIDADE DE SOFTWARE

Qualidade de software trata-se de uma das áreas da engenharia de software que objetiva garantir a qualidade do produto através do processo de desenvolvimento definido (PERUCCI; CAMPOS, 2016). É um processo metódico, focado nos artefatos e em todas as etapas desenvolvidas, que busca garantir a conformidade de produtos e processos, evitando ou eliminando possíveis defeitos. (SAMBO, 2017).

A qualidade do software está diretamente ligada à qualidade do processo de seu desenvolvimento. Sua avaliação refere-se a um processo subjetivo por meio do qual é preciso verificar se o software corresponde ou não a sua finalidade diante das suas características (SOMMERVILLE, 2018). Dessa forma é importante que as organizações definam quais as técnicas, padrões, modelos e procedimentos a serem seguidos, para que possam então buscar desenvolver um software de qualidade (SILVA; FIGUEIREDO, 2019)

Na garantia da qualidade do software, seus atributos devem ser satisfeitos de forma a atender as necessidades dos envolvidos. Estes são definidos de acordo com a sua relevância para o software em questão: tecnologia utilizada, características específicas do projeto, necessidades da organização e do usuário e domínio da aplicação (ANDRADE, 2015).

Além da implementação correta das funcionalidades do software, sua qualidade depende dos atributos não funcionais. Os atributos mais importantes de qualidade do software referem-se a sua eficiência, confiabilidade, manutenção e segurança. (SOMMERVILLE, 2018). A qualidade do software também depende da concepção dos responsáveis por sua avaliação: desenvolvedor, usuário e organização, os quais avaliam, respectivamente, os aspectos de conformidade a partir dos fatores internos e requisitos do cliente, os aspectos internos focados no desempenho, custo, facilidade de uso e confiabilidade dos resultados, bem como os aspectos de custo, cronograma e conformidade, de acordo com os requisitos estabelecidos por clientes e desenvolvedores (ANDRADE, 2015).

2.2.1 Processo de Garantia da Qualidade de Software

A garantia da qualidade de software (SQA, *software quality assurance*) é um conjunto de atividades de apoio, as quais passam segurança de que os processos foram definidos e são otimizados constantemente, para o desenvolvimento de produtos que sigam suas especificações e que cumpram com os objetivos estabelecidos (SAMBO, 2017).

Apesar do processo ser um fator primordial na garantia da qualidade do produto, é importante que os gerentes de qualidade estabeleçam a sua própria cultura, como afirma Sommerville (2018):

Os gerentes da qualidade devem também ter como objetivo desenvolver uma 'cultura da qualidade', na qual todos os responsáveis pelo desenvolvimento de software estão comprometidos em atingir um alto nível de qualidade de produto. Eles devem incentivar os times a assumir a responsabilidade pela qualidade do seu trabalho e a desenvolver novas abordagens de melhoria da qualidade. Embora os padrões e os procedimentos sejam a base do gerenciamento da qualidade, os bons gerentes da qualidade reconhecem que existem aspectos intangíveis para a qualidade do software (elegância, legibilidade etc.) que não podem ser incorporados a padrões."

De acordo com Pressman e Maxim (2016), o processo de garantia de qualidade de software envolve uma série de atividades a serem seguidas pela SQA, sendo estas:

Padrões: ao definir os padrões de desenvolvimento, a SQA precisa garantir que estes padrões foram seguidos e que os produtos finais encontram-se em conformidade com eles.

Revisões e auditorias: atividades de qualidade realizadas entre engenheiros de software com o objetivo de expor erros. A SQA realiza estas atividades como revisões que venham a assegurar que as diretrizes de qualidade estão sendo seguidas.

Testes: atividade de controle de qualidade, com o objetivo de encontrar erros. É necessário que a SQA garanta que os testes sejam devidamente planejados e conduzidos de forma correta, com a maior probabilidade possível de atingir seu principal objetivo.

Coleta e análise de erros/defeitos: para melhor compreender como ocorre a introdução dos erros e quais são as atividades mais adequadas da engenharia de software, para sua eliminação, a SQA reúne e faz uma análise de dados de erros e defeitos.

Gerenciamento de mudanças: a SQA garante que sejam instituídas práticas de gerenciamento de mudanças adequadas, que não comprometam a qualidade do software.

Educação: fator primordial no aperfeiçoamento dos envolvidos no processo de desenvolvimento de software. A SQA é líder no processo de aperfeiçoamento

do software, sendo um proponente fundamental e patrocinador de programas educacionais.

Gerência de fornecedores: a SQA precisa garantir um software de alta qualidade, através de práticas específicas da garantia de qualidade de software, que o fornecedor tem de seguir, e como parte de qualquer contrato com fornecedores externos, exigir condições de qualidade.

Administração da segurança: é necessário que sejam instituídas políticas de segurança por qualquer organização de software, garantindo que o software não sofra alterações internas sem autorização. A SQA garante a aplicação de processos e tecnologias para que o software possua a segurança desejada.

Proteção: a SQA pode se responsabilizar pela avaliação dos impactos de falhas do software e dar início às etapas necessárias na diminuição de riscos.

Gestão de riscos: a SQA garante que planos de contingência, referentes a riscos, foram estabelecidos e que as atividades de gestão de riscos sejam conduzidas de forma adequada.

É possível alcançar a garantia de qualidade do software por meio de ferramentas que controlam as técnicas e padrões existentes referente a qualidade do software, assim a SQA possui a função de analisar se os padrões, técnicas e métodos, definidos no decorrer da engenharia de software, estão sendo devidamente aplicados pelos profissionais envolvidos (SANTOS; OLIVEIRA, 2017).

2.3 TESTE DE SOFTWARE

O teste de software é uma atividade fundamental na garantia de qualidade do software, por meio da qual é possível identificar se o produto a ser entregue segue as especificações estabelecidas (GÉRGOLI, 2017). Trata-se de um método de validação e execução controlada de um programa, que tem como objetivo avaliar o seu desempenho (TRIGO, 2017).

O foco das atividades de teste está em evitar, prevenir e encontrar falhas durante o processo de software, o que requer o conhecimento de métodos e técnicas de testes específicos (FERNANDES, 2017).

Os testes trabalham na perspectiva dos processos de verificação e validação de software, que têm como objetivo estabelecer a segurança de que o software adequa-se a

sua finalidade, sendo responsáveis, respectivamente, por avaliar se o software atende aos requisitos (funcionais e não funcionais) especificados e assegurar que o software corresponde às expectativas do cliente, funcionando como o esperado (SOMMERVILLE, 2018).

Quando capazes de aumentar a qualidade da entrega do produto final de forma expressiva, o teste de software é considerado significativo e bem elaborado, caso contrário, uma aplicação testada indevidamente pode gerar graves problemas, tais como prejuízos financeiros e sua recodificação total (SILVA; DANILLO, 2019).

Para garantir que as atividades de testes estão sendo desempenhadas corretamente é importante que as especificações dos testes sejam revisadas antes de serem postas em prática, para isso é necessário a elaboração de planos e procedimentos eficazes, por meio dos quais será possível o desenvolvimento ordenado do software e a descoberta de erros no decorrer do processo (PRESSMAN; MAXIM, 2016).

2.3.1 Técnicas de Teste

Para que os testes de software venham a avaliar todos os aspectos do software existem diferentes técnicas de testes que devem ser aplicadas no processo de avaliação (FERNANDES, 2017). Estas técnicas oferecem métodos para o planejamento de testes que auxiliam na descoberta de erros relacionados ao funcionamento, desempenho e comportamento do software (PRESSMAN; MAXIM, 2016).

As principais técnicas de testes são: funcional e estrutural, conhecidas respectivamente como teste de caixa preta e teste de caixa branca, e são indispensáveis em qualquer planejamento de testes, a diferença entre elas está no nível e em como os testes são realizados (PRIMÃO A. P; RIBEIRO, 2015).

2.3.1.1 Teste Funcional

O teste funcional é focado nos requisitos não funcionais, documentados a partir da especificação do software (FERNANDES, 2017). Nesse processo, a implementação do código não é avaliada, há a comparação entre a saída resultante da entrada de dados com a saída que se espera, conforme os requisitos (BARBIERI, 2018).

A principal vantagem da utilização desta técnica está no fato de que o responsável por sua realização não precisa ter conhecimento a respeito da complexidade dos conceitos utilizados na parte interna do software ou das tecnologias empregadas (CAUZZI, 2015).

Segundo Pressman e Maxim (2016), os testes funcionais descartam a estrutura de controle, voltando sua atenção ao domínio das informações e seu objetivo é detectar erros que abordem as seguintes categorias:

- Ausência ou erro de funções;
- Erros comportamentais e de desempenho;
- Erros de interface;
- Erros estruturais dos dados ou acesso a base de dados externas;
- Erros de inicialização e término.

2.3.1.2 Teste Estrutural

O teste estrutural consiste na estrutura interna do programa e em seus componentes, tendo como foco principal seu código-fonte (SOMMERVILLE, 2018). Todos os seus componentes são classificados de acordo com a complexidade do sistema, assim como o fluxo de controle e de dados. (SOUZA et al., 2017).

Para que esta técnica seja aplicada é necessário que o responsável conheça a tecnologia utilizada no desenvolvimento do software para que os caminhos a serem realizados sejam determinados, assim estabelecendo parâmetros de entrada e os resultados desejados (CAUZZI, 2015).

De acordo com Pressman e Maxim (2016) a partir do teste estrutural é possível que sejam criados casos de testes que exercitem os seguintes parâmetros:

- Todos os caminhos independentes do módulo;
- Todas as decisões lógicas;
- Todos os ciclos;
- Estrutura de dados interna.

2.3.2 Níveis de teste

Como visto anteriormente na Figura 3, o teste de software divide-se em quatro níveis: teste de unidade, teste de integração, teste de sistema e teste de aceitação (PRESSMAN; MAXIM, 2016). Os quais serão abordados nesta seção.

2.3.2.1 Teste de Unidade

O teste de unidade é um processo de testagem dos componentes mais simples do sistema, dos quais podem ser citados os métodos e as classes (SOMMERVILLE, 2018). Estes componentes são analisados pelo desenvolvedor do código, individualmente, de acordo com o desenvolvimento da unidade, com o intuito de identificar falhas decorrentes de defeitos de lógica (FERNANDES, 2017).

2.3.2.2 Teste de Integração

O teste de integração verifica se todas as etapas do programa, quando integradas, funcionam de forma correta (NOELLO, 2016). Trata-se de uma técnica sistemática na construção da arquitetura do software, além de realizar testes que possam identificar possíveis erros de interface, com a finalidade de desenvolver a estrutura do programa estabelecida pelo projeto a partir dos componentes testados em unidade (PRESSMAN; MAXIM, 2016).

2.3.2.3 Teste de Sistema

O teste de sistema tem como objetivo comparar as características especificadas durante o projeto com as características encontradas do programa, esse processo ocorre após os módulos serem integrados (CAMARA, 2016). Durante esse processo é verificado se os componentes são compatíveis, interagem de forma correta ou se a transferência de dados ocorre no momento certo, através de suas interfaces (SOMMERVILLE, 2018).

2.3.2.4 Teste de Aceitação

O objetivo do teste de aceitação é verificar se o software corresponde aos requisitos especificados (BARBIERI, 2018). Este processo é realizado antes que o software seja entregue, geralmente é conduzido por um grupo de usuários finais por meio de simulações e pode ser classificado como formal, quando há o gerenciamento dos casos de testes e como informal, quando os testes não são gerenciados (FERNANDES, 2017).

2.3.3 Processo de Teste de Software

No processo de desenvolvimento de software é importante que as etapas destinadas aos testes sejam definidas (GÉRGOLI, 2017). O processo de teste abrange quatro etapas que devem ser realizadas no decorrer de toda atividade: planejamento, casos de teste, execução e avaliação de acordo com Delamaro, Maldonado e Jino (2016 apud SOUZA et al., 2017).

2.3.3.1 Planejamento de Teste

O planejamento de teste busca atender os requisitos de qualidade especificados pelo cliente (FERNANDES, 2017). Sua preocupação está em definir os recursos que serão utilizados durante todas as atividades de teste, partindo do processo estabelecido e das pessoas envolvidas, assim como a disponibilidade de tempo, para isso é desenvolvido um plano de testes, por meio do qual são definidos o cronograma de realização, os testes a serem realizados e a forma como eles serão registrados (SOMMERVILLE, 2018).

2.3.3.2 Casos de Teste

Os casos de teste referem-se às especificações das entradas para os testes e os seus resultados (SOMMERVILLE, 2018). Eles visam a identificação de todos os testes possíveis, os quais podem ser identificados por meio da análise dos requisitos e estrutura interna, e/ou por meio de técnicas de inspeção de documentos (PRIMÃO A. P; RIBEIRO, 2015) .

Segundo Gérgoli (2017), para que os casos de testes sejam realizados é necessário responder às seguintes questões: O que se pretende testar? Quando será testado? Como será testado? Onde será testado?

2.3.3.3 Execução do Teste

No processo de execução os casos de testes passam a ser executados por meio da simulação da ação dos usuários no sistema (FERNANDES, 2017). A execução do teste deve estar associada aos planos de testes de integração ou testes de sistemas, a execução desses testes deve ser realizada por requisitos e release, respectivamente (GÉRGOLI, 2017).

Os testes podem ser executados por meio de duas abordagens: manual e automatizada (FERNANDES, 2017).

Na abordagem manual o teste é realizado por um testador que compara as informações do teste com o resultado esperado e ao encontrar falhas, reporta as informações para o desenvolvedor. Enquanto na abordagem automatizada, há a codificação do teste em programa, em que a cada vez que o software é testado ele é executado, o que faz da abordagem automatizada mais rápida do que a abordagem manual (SALOMÃO, 2016).

2.3.3.4 Avaliação dos Resultados de Teste

Nesta etapa ocorre a análise das falhas encontradas e de seus impactos, originando relatórios e indicadores. Quando o teste não consegue detectar as falhas, ocorre a revisão dos casos de testes, a partir da qual é decidido se a atividade de teste será, ou não, finalizada (FERNANDES, 2017).

Com base na literatura apresentada, o presente trabalho abordará a realização das atividades de testes em algumas empresas desenvolvedoras de software.

Para isso foram utilizados alguns métodos e ferramentas que serão apresentados no próximo capítulo.

3 PROCEDIMENTOS METODOLÓGICOS

3.1 CARACTERIZAÇÃO DO OBJETO DE ESTUDO

A presente pesquisa contou com a participação de cinco empresas desenvolvedoras de softwares do âmbito privado, as quais, por questões de confidencialidade, não serão identificadas. Estas empresas estão situadas no estado do Rio Grande do Norte, especificamente nas cidades de Caicó, Natal e Nova Cruz e se tratam de empresas de grande, médio e pequeno porte.

3.2 MÉTODO DA PESQUISA

O estudo em questão busca analisar e compreender a aplicabilidade e a importância das atividades de testes de software em empresas privadas. Diante disso, a natureza do trabalho caracteriza-se como descritiva, que permite analisar, interpretar, registrar e estudar fatos que podem ser comprovados (CASTILHO; BORGES; PEREIRA, 2011).

No tratamento e interpretação dos dados utilizou-se o método quantitativo, através do qual as informações são apresentadas por meio de procedimentos estatísticos para a análise e classificação dos dados, sendo necessário a formulação de hipóteses e a exposição das relações existentes entre as variáveis, garantindo a precisão dos resultados (PRODANOV; FREITAS, 2013).

Para coleta de informações foi realizada aplicação de um questionário adaptado da proposta de Neto et al. (2006), do qual foram utilizadas vinte e uma questões e acrescentada uma. As questões abordam: os tipos, organização, planejamento, controle, medição, análise e ferramentas utilizadas durante a realização das atividades de testes.

As informações são avaliadas mediante a sua aplicabilidade e importância, seguindo uma escala que varia de 0 a 4, representando um aumento gradativo das informações como mostra a tabela 1.

Tabela 1 – Descrição da Escala de Aplicabilidade e Importância do Questionário Aplicado

APLICABILIDADE	IMPORTÂNCIA
(0) Não aplicável	(0) Não importante
(1) Não utilizado pela organização	(1) Pouco importante
(2) Uso não frequente	(2) Valor limitado: poderia ser adequado utilizá-lo
(3) Uso habitual: usado em grande parte dos projetos de software da organização	(3) Valor signficante: prática recomendada
(4) Uso padrão: usado em todos os projetos de software da organização	(4) Valor essencial: deve ser uma prática padrão para todos

Fonte: Adaptado de Neto et al. (2006)

A aplicação do questionário ocorreu entre o período de 14 a 23 de fevereiro de 2022, através da ferramenta Google Forms, o qual foi enviado por link para os participantes através de seus contatos (E-mail e WhatsApp).

3.3 APRESENTAÇÃO E ANÁLISE DE DADOS

A análise dos dados se deu a partir da questão dois (Q2), pois a primeira questão não fez uso de escala para a obtenção de respostas, objetivando apenas saber se as empresas realizam ou não o teste de software.

Ao responderem o questionário, todas as empresas afirmaram realizar o teste de software. Tendo em vista que o questionário busca obter os dados de acordo com a aplicabilidade e importância das atividades de testes de software, a análise ocorreu em três etapas: análise baseada na aplicabilidade, análise baseada na importância e por fim a correlação entre ambas. Para isso foram utilizadas as seguintes funções estatísticas: máximo, mínimo, média e desvio padrão, que foram calculados a partir da tabulação dos dados no Excel.

3.3.1 Análise Baseada na Aplicabilidade

A tabela 2 apresenta o percentual das respostas obtidas para cada variável, quanto a sua aplicabilidade, na qual os maiores percentuais estão destacados.

Tabela 2 – Percentual das Respostas Obtidas Para Cada Variável, Baseado na Aplicabilidade

Questões	Não aplicável (0)	Não utilizado (1)	Uso não frequente (2)	Uso habitual (3)	Uso padrão (4)
Q2 - Existência de um responsável ou uma equipe alocada para as atividades de testes.	0%	0%	20%	40%	40%
Q3 - Realização de teste de Unidade.	0%	20%	60%	0%	20%
Q4 - Realização de teste de Integração.	0%	40%	40%	20%	0%
Q5 - Realização de teste de Sistema.	0%	0%	0%	20%	80%
Q6 - Realização de teste de Aceitação.	0%	20%	60%	0%	20%
Q7 - Realização de treinamentos sobre testes.	0%	0%	60%	20%	20%
Q8- Separação das atividades de teste do desenvolvimento .	0%	0%	60%	20%	20%
Q9- Existência de testadores em tempo integral para a realização dos testes.	40%	0%	0%	20%	40%
Q10 - Utilização de metodologia ou processo para sistematizar e organizar os testes.	0%	20%	40%	20%	20%

Tabela 2 – Continuação

Questões	Não aplicável (0)	Não utilizado (1)	Uso não frequente (2)	Uso habitual (3)	Uso padrão
Q11- Re-execução dos testes quando o software é modificado.	0%	0%	40%	20%	40%
Q12 - Documentação do plano de teste descrevendo objetivos/abordagens.	40%	0%	60%	0%	0%
Q13 - Documentação de caso de testes e resultados.	20%	60%	20%	0%	0%
Q14 - Identificação e utilização de riscos, para projetar, organizar e executar os testes de software.	20%	60%	20%	0%	0%
Q15 - Desenvolvimento de teste antes da codificação	60%	40%	0%	0%	0%
Q16 - Registro do tempo gasto em testes.	40%	40%	0%	0%	20%
Q17 - Registro de falhas e defeitos detectados durante os testes.	0%	0%	40%	40%	20%
Q18 - Conservação dos testes para usos futuros.	40%	0%	40%	20%	0%

Tabela 2 – Continuação

Questões	Não aplicável (0)	Não utilizado (1)	Uso não frequente (2)	Uso habitual (3)	Uso padrão
Q19 - Análise de defeitos identificados durante testes para descobrir suas causas e como foram introduzidos.	0%	0%	40%	60%	0%
Q20 - Utilização de ferramentas geradoras de procedimentos/casos de testes.	0%	40%	40%	0%	20%
Q21 - Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste.	20%	0%	40%	20%	20%
Q22 - Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.	20%	20%	40%	20%	0%

Fonte: Dados da pesquisa, 2022

Além dos percentuais, também foram calculados, o valor máximo, o valor mínimo, a média e o desvio padrão de cada variável, como mostra a tabela 3.

Tabela 3 – Análise Estatística dos Dados Referente a Aplicabilidade

Questões	Max	Min	Média	Desvio padrão
Q2 - Existência de um responsável ou uma equipe alocada para as atividades de testes	4	2	3,2	0,75
Q3 - Realização de teste de Unidade.	4	1	2,2	0,98
Q4 - Realização de teste de Integração.	3	1	1,8	0,75
Q5 - Realização de teste de Sistema.	4	3	3,8	0,40
Q6 - Realização de teste de Aceitação.	4	1	2,2	0,98
Q7 - Realização de treinamentos sobre testes.	4	2	2,6	0,80
Q8- Separação das atividades de teste do desenvolvimento.	4	2	2,6	0,80
Q9- Existência de testadores em tempo integral para a realização dos testes.	4	0	2,2	1,83
Q10 - Utilização de metodologia ou processo para sistematizar e organizar os testes.	4	1	2,4	1,02
Q11- Re-execução dos testes quando o software modificado.	4	2	3	0,89
Q12 - Documentação do plano de teste descrevendo objetivos/abordagens.	2	0	1,2	0,98
Q13 - Documentação de caso de testes e resultados.	2	0	1	0,63
Q14 - Identificação e utilização de riscos, para projetar, organizar e executar os testes de software.	2	0	1	0,63
Q15 - Desenvolvimento de teste antes da codificação	1	0	0,4	0,49
Q16 - Registro do tempo gasto em testes.	4	0	1,2	1,47
Q17 - Registro de falhas e defeitos detectados durante os testes.	4	2	2,8	0,75
Q18 - Conservação dos testes para usos futuros.	3	0	1,4	1,20
Q19 - Análise de defeitos identificados durante testes para descobrir suas causas e como foram introduzidos.	3	2	2,6	0,49

Tabela 3 – Continuação

Questões	Max	Min	Média	Desvio padrão
Q20 - Utilização de ferramentas geradoras de procedimentos/casos de testes.	4	1	2	1,10
Q21 - Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste.	4	0	2,2	1,33
Q22 -Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.	3	0	1,6	1,02

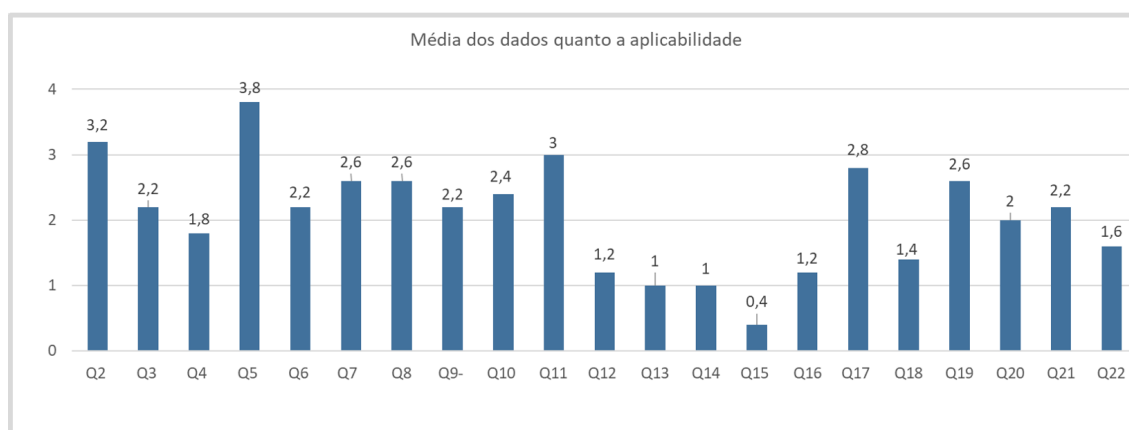
Fonte: Dados da pesquisa, 2022

A partir dos dados estatísticos apresentados na tabela 3, é possível observar que o valor máximo que aparece com maior frequência é o 4, e quanto ao indicador mínimo, o valor predominante é o 0, estes indicadores correspondem respectivamente às variáveis definidas como “uso padrão” e “não aplicável”.

No que se refere a média das variáveis para cada questão, os resultados obtidos ficaram entre 0,4 e 3,8. Com isso nota-se que as questões Q2, Q5 e Q11 apresentam as maiores médias: 3,2, 3,8, e 3,0 respectivamente, as quais, com base da escala utilizada, a Q5 por se aproximar do valor 4 corresponde a variável de uso padrão e as Q2 e Q11, por se aproximarem do valor 3, correspondem a variável de uso habitual.

Ainda sobre as médias obtidas, as questões Q12, Q13, Q14, Q15, Q16 e Q18 apresentam as menores médias, das quais a Q15 obteve a média mais baixa de todas as questões resultando em 0,4, que por se aproximar de 0 indica que estas atividades não se aplicam às empresas, e as Q13, Q14, Q16 e Q18 apresentaram uma média aproximada de um, o que significa que não são utilizadas pelas empresas no processo de testes.. A figura 6 apresenta o resumo das médias obtidas.

Figura 6 – Média dos Dados Quanto a Aplicabilidade



Fonte: Dados da pesquisa, 2022

Quanto ao desvio padrão obtido, a questão que apresentou maior índice de divergência entre as respostas foi a Q9 com um valor de 1,83.

Com base na análise realizada a partir da aplicabilidade das atividades de testes de software, foi constatado que:

- O teste de sistema é realizado, por grande parte das empresas, em todos os seus projetos (Q5);
- Há um responsável pela realização das atividades de testes em boa parte dos projetos realizados (Q2);
- Ao haver mudanças em um determinado software, é a realizada a reexecução dos testes(Q11);
- A realização de testes antes da codificação não se aplica ao processo de desenvolvimento (Q15);
- Os casos de testes não são documentados(Q13);
- Não há o registro do tempo de realização de testes (Q16);
- Não há a conservação de testes para usos futuros (Q18);
- Não há a análise dos defeitos identificados no processo de teste (Q19);

Quanto às demais atividades avaliadas, elas estão entre as não aplicáveis ou não aplicadas frequentemente.

3.3.2 Análise Baseada na Importância

A tabela 4 apresenta o percentual das respostas obtidas para cada variável, quanto a sua importância, na qual os maiores percentuais estão destacados.

Tabela 4 – Percentual das Respostas Obtidas Para Cada Variável, Baseado na Importância

Questões	Não importante (0)	Pouco importante (1)	Valor limitado (2)	Valor significativo (3)	Valor Essencial (4)
Q2 - Existência de um responsável ou uma equipe alocada para as atividades de testes	0%	0%	20%	0%	80%
Q3 - Realização de teste de Unidade.	0%	0%	40%	0%	60%
Q4 - Realização de teste de Integração	0%	20%	20%	40%	20%
Q5 - Realização de teste de Sistema.	0%	0%	0%	40%	60%
Q6 - Realização de teste de Aceitação.	0%	20%	20%	40%	20%
Q7 - Realização de treinamentos sobre testes.	0%	0%	20%	80%	0%
Q8- Separação das atividades de teste do desenvolvimento.	0%	0%	40%	40%	20%

Tabela 4 – Continuação

Questões	Não importante (0)	Pouco importante (1)	Valor limitado (2)	Valor significativo (3)	Valor Essencial (4)
Q9- Existência de testadores em tempo integral para a realização dos testes.	40%	0%	0%	20%	40%
Q10 - Utilização de metodologia ou processo para sistematizar e organizar os testes.	0%	0%	0%	80%	20%
Q11- Re-execução dos testes quando o software é modificado.	0%	0%	0%	40%	60%
Q12 - Documentação do plano de teste descrevendo objetivos/abordagens.	20%	0%	20%	60%	0%
Q13 - Documentação de caso de testes e resultados.	20%	20%	40%	20%	0%
Q14 - Identificação e utilização de riscos, para projetar, organizar e executar os testes de software.	0%	40%	20%	40%	0%
Q15 - Desenvolvimento de teste antes da codificação	40%	0%	40%	20%	0%

Tabela 4 – Continuação

Questões	Não importante (0)	Pouco importante (1)	Valor limitado (2)	Valor significativo (3)	Valor Essencial (4)
Q16 - Registro do tempo gasto em testes.	20%	0%	60%	0%	20%
Q17 - Registro de falhas e defeitos detectados durante os testes.	0%	0%	40%	60%	0%
Q18 - Conservação dos testes para usos futuros.	20%	0%	60%	20%	0%
Q19 - Análise de defeitos identificados durante testes para descobrir suas causas e como foram introduzidos.	0%	0%	0%	100%	0%
Q20 - Utilização de ferramentas geradoras de procedimentos/casos de testes.	0%	0%	40%	20%	40%
Q21 - Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste.	0%	0%	20%	60%	20%

Tabela 4 – Continuação

Questões	Não importante (0)	Pouco importante (1)	Valor limitado (2)	Valor significativo (3)	Valor Essencial (4)
Q22 -Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.	0%	0%	60%	40%	0%

Fonte: Dados da pesquisa, 2022

Além dos percentuais, foram calculados o valor máximo, o valor mínimo, a média e o desvio padrão de cada variável, como mostra a tabela 5.

Tabela 5 – Análise Estatística dos Dados Referente a Importância

Questões	Max	Min	Média	Desvio padrão
Q2 - Existência de um responsável ou uma equipe alocada para as atividades de testes	4	2	3,6	0,89
Q3 - Realização de teste de Unidade.	4	2	3,2	1,10
Q4 - Realização de teste de Integração.	4	1	2,6	1,14
Q5 - Realização de teste de Sistema.	4	3	3,6	0,55
Q6 - Realização de teste de Aceitação.	4	1	2,6	1,14
Q7 - Realização de treinamentos sobre testes.	3	2	2,8	0,45
Q8- Separação das atividades de teste do desenvolvimento.	4	2	2,8	0,84
Q9- Existência de testadores em tempo integral para a realização dos testes.	4	0	2,8	1,64

Tabela 5 – Continuação

Questões	Max	Min	Média	Desvio padrão
Q10 - Utilização de metodologia ou processo para sistematizar e organizar os testes.	4	3	3,2	0,45
Q11- Re-execução dos testes quando o software é modificado.	4	3	3,6	0,55
Q12 -Documentação do plano de teste descrevendo objetivos/abordagens.	3	0	2,2	1,30
Q13 - Documentação de caso de testes e resultados.	3	0	1,6	1,14
Q14 - Identificação e utilização de riscos, para projetar, organizar e executar os testes de software.	3	1	2	1,00
Q15 - Desenvolvimento de teste antes da codificação	3	0	1,4	1,34
Q16 - Registro do tempo gasto em testes.	4	0	2	1,41
Q17 - Registro de falhas e defeitos detectados durante os testes.	3	2	2,6	0,55
Q18 - Conservação dos testes para usos futuros.	3	0	1,8	1,10
Q19 - Análise de defeitos identificados durante testes para descobrir suas causas e como foram introduzidos.	3	3	3	0,00
Q20 - Utilização de ferramentas geradoras de procedimentos/casos de testes.	4	2	3	1,00
Q21 - Utilização de ferramentas de gerência de teste para rastrear registrar os resultados da execução de teste.	4	2	3	0,71

Tabela 5 – Continuação

Questões	Max	Min	Média	Desvio padrão
Q22 - Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.	3	2	2,4	0,55

Fonte: Dados da pesquisa, 2022.

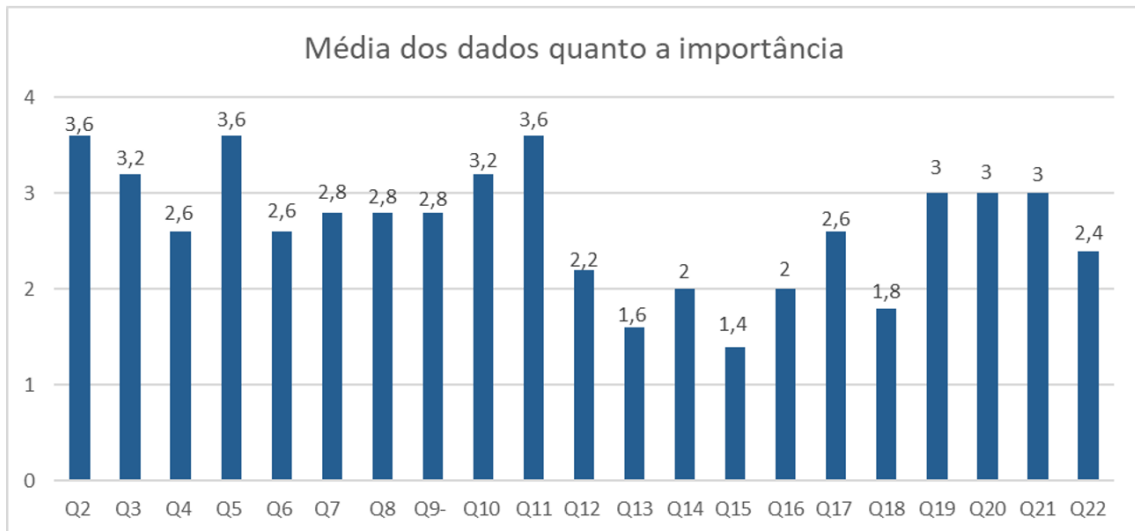
Com base nos dados estatísticos apresentados na tabela 5, nota-se que o valor máximo que aparece com mais frequência é o 4, correspondendo a variável que indica que as atividades possuem um valor essencial quanto a sua importância, já o valor mínimo mais frequente é o 2, o que indica que as atividades possuem um valor de importância limitado.

Quanto às médias obtidas, seus valores variam de 1,4 a 3,6. As questões de maior média são Q2, Q5 e Q11, ambas com um valor de 3,6, que por se aproximarem de 4,0 tratam-se de atividades essenciais. Já as questões que apresentam menor média são as Q13, Q15 e a Q18, as quais não são consideradas muito importantes. As demais questões encontram-se entre uma média de 2 a 3, julgadas com um valor de importância limitado (quando próximo de 2) ou significativa (quando próximo de 3). O resumo das médias obtidas é apresentado na figura 7.

Quanto ao desvio padrão obtido, a questão que apresentou maior índice de divergência entre as respostas foi a Q9 com um valor de 1,64. A partir da análise realizada de acordo com a importância das atividades de testes de software, foi constatado que:

- É essencial a existência de um responsável pela realização dos testes (Q2);
- A realização dos testes de sistemas é uma atividade essencial(Q5);
- É essencial que haja a reexecução de testes diante alguma modificação do software (Q11);
- A documentação dos casos de testes não é considerada de muita importância(Q13);
- Há pouca importância quanto a atividade de identificação e utilização dos riscos(Q14);

Figura 7 – Média dos Dados Quanto a Importância



Fonte: Dados da pesquisa, 2022

- Há pouca importância quanto a realização de testes antes da codificação do software(Q15).

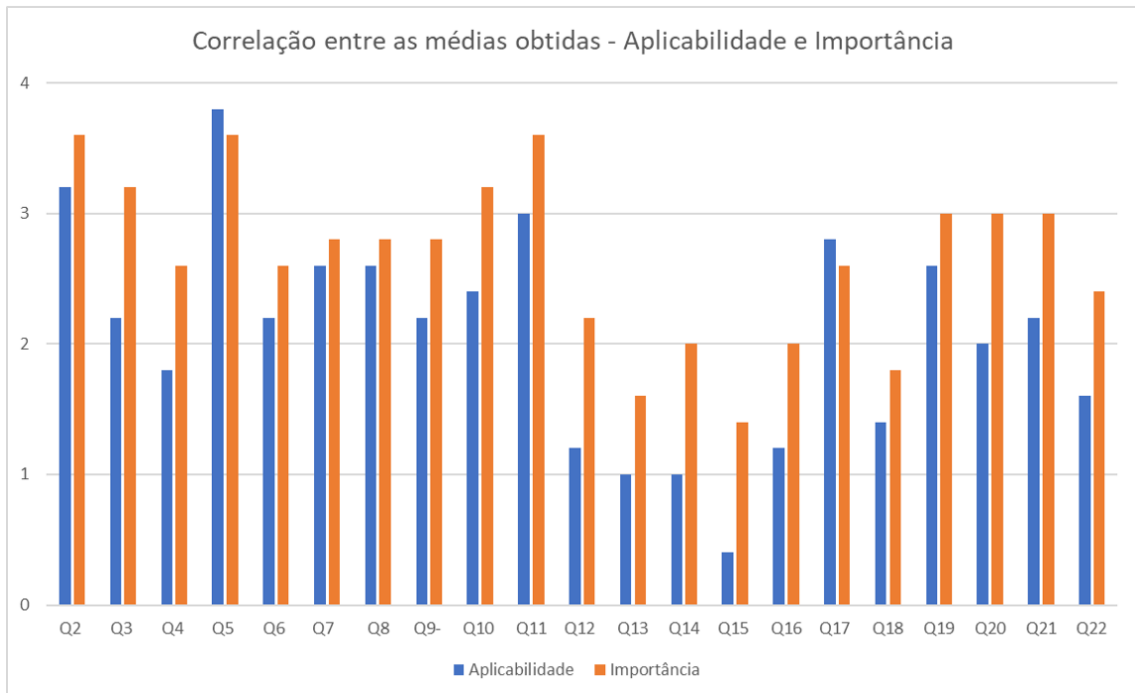
Quanto às demais atividades analisadas, as quais obtiveram uma média entre 2,0 e 3,0 apresentam valor limitado e significativo, ou seja, são atividades adequadas e recomendadas no processo de teste, respectivamente.

3.3.3 Correlação Entre os Resultados Obtidos

Após a análise dos dados quanto a sua aplicabilidade e importância realizadas separadamente, foi feita a correlação entre os resultados obtidos.

Com análises estatísticas realizadas foi possível observar que os resultados obtidos, quanto às médias, indicam que as questões Q2, Q5 e Q11 referentes, respectivamente, a existência de um responsável para a realização de testes, a realização de teste de sistema e a reexecução das atividades de testes quando o software passar por modificações, obtiveram as maiores médias tanto em aplicabilidade como em importância, as questões Q13, Q14 e Q15 apresentaram as menores médias também em ambos os fatores analisados, as quais referem-se a documentação dos casos de testes e resultados, identificação de riscos para a realização dos testes e a realização de testes antes da codificação, respectivamente, o que é possível observar na figura 8.

Figura 8 – Correlação Entre as Médias Obtidas: Aplicabilidade e Importância



Fonte: Dados da pesquisa, 2022

A tabela 6 expõe a relação entre os resultados de aplicabilidade e importância para cada atividade, conforme as variáveis utilizadas na escala de avaliação.

Tabela 6 – Relação Entre os Resultados Obtidos: Aplicabilidade e Importância

Questões	Aplicabilidade	Importância
Q2 - Existência de um responsável ou uma equipe alocada para as atividades de testes	Uso habitual	Valor essencial
Q3 - Realização de teste de Unidade.	Uso não frequente	Valor significativo
Q4 - Realização de teste de Integração.	Uso não frequente	Valor significativo
Q5 - Realização de teste de Sistema.	Uso padrão	Valor essencial
Q6 - Realização de teste de Aceitação.	Uso não frequente	Valor significativo

Tabela 6 – Continuação

Questões	Aplicabilidade	Importância
Q7 - Realização de treinamentos sobre testes.	Uso habitual	Valor significativo
Q8- Separação das atividades de teste do desenvolvimento.	Uso habitual	Valor significativo
Q9- Existência de testadores em tempo integral para a realização dos testes.	Uso não frequente	Valor significativo
Q10 - Utilização de metodologia ou processo para sistematizar e organizar os testes.	Uso não frequente	Valor significativo
Q11- Re-execução dos testes quando o software é modificado.	Uso habitual	Valor essencial
Q12 -Documentação do plano de teste descrevendo objetivos/abordagens.	Não utilizado	Valor limitado
Q13 - Documentação de caso de testes e resultados.	Não utilizado	Pouco importante
Q14 - Identificação e utilização de riscos, para projetar, organizar e executar os testes de software.	Não utilizado	Valor limitado
Q15 - Desenvolvimento de teste antes da codificação	Não aplicável	Pouco importante
Q16 - Registro do tempo gasto em testes.	Não utilizado	Valor limitado
Q17 - Registro de falhas e defeitos detectados durante os testes.	Uso habitual	Valor significativo
Q18 - Conservação dos testes para usos futuros.	Não utilizado	Valor limitado
Q19 - Análise de defeitos identificados durante testes para descobrir suas causas e como foram introduzidos.	Uso habitual	Valor significativo

Tabela 6 – Continuação

Questões	Aplicabilidade	Importância
Q20 - Utilização de ferramentas geradoras de procedimentos/casos de testes.	Uso não frequente	Valor significativo
Q21 - Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste.	Uso não frequente	Valor significativo
Q22 - Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.	Uso não frequente	Valor limitado

Fonte: Dados da pesquisa, 2022

A partir das informações da tabela 6, nota-se que das vinte e uma atividades de testes avaliadas, seis não são aplicadas pelas empresas participantes do estudo, sendo estas:

- Documentação do plano de testes, descrevendo objetivos e abordagens (Q12);
- Documentação de casos de testes e resultados (Q13);
- Identificação e utilização de riscos para projetar, organizar e executar os testes de software (Q14);
- Desenvolvimento de testes antes da codificação (Q15);
- Registro de tempo de teste (Q16);
- Conservação de testes para usos futuros (Q80).

Quanto à importância, é possível observar que apesar de que algumas atividades não sejam aplicadas, todas são consideradas importantes no processo de desenvolvimento do software, mesmo que algumas apresentem pouca importância (Q13 e Q15). Nota-se também que as atividades que possuem valor essencial, quanto a escala utilizada são:

- A existência de um responsável ou equipe alocada para a realização dos testes (Q2);

- Realização de testes de sistema (Q5);
- Reexecução dos testes quando há mudanças no software (Q11).

Portanto, no contexto analisado, conclui-se que apesar das empresas envolvidas realizarem o de testes de software, nota-se que não há a realização e documentação de planos e processos de testes realizados, identificação de riscos que contribuam na organização e execução dos testes, nem a conservação dos testes realizados.

Estas atividades são indispensáveis, pois a partir delas é possível a elaboração de um plano de testes, o qual permite a identificação das atividades e recursos a serem utilizados, evitando uma aplicação indevida dos testes, o que, segundo Silva e Danillo (2019), pode causar problemas que levem a prejuízos financeiros e de codificação do sistema, comprometendo, assim, a qualidade do produto.

4 CONSIDERAÇÕES FINAIS

A qualidade de software liga-se diretamente ao seu processo de desenvolvimento. Esta qualidade possui uma influência significativa diante a competitividade do mercado de desenvolvimento de software, fazendo com que as empresas desenvolvedoras busquem estabelecer um processo por meio do qual seja possível alcançar a qualidade de seu produto. Neste processo, o teste de software é indispensável, pois sua aplicação permite identificar se o software corresponde a sua finalidade.

Diante deste contexto, o presente trabalho objetivou analisar as atividades de testes desempenhadas em empresas do ramo privado quanto à sua aplicabilidade e importância. O questionário utilizado para a obtenção dos dados apresentou eficácia quanto ao alcance do objetivo estabelecido, por meio do qual foi possível constatar que as empresas participantes afirmam realizar o teste de software no processo de desenvolvimento.

No entanto, algumas das atividades de testes apesar de serem consideradas importantes no processo de desenvolvimento, não são aplicadas pelas empresas, tais como as atividades voltadas para a documentação do processo de testes. A não realização dessas atividades pode causar prejuízos financeiros e de codificação, comprometendo assim a qualidade do produto final.

Com isso, a pesquisa realizada foi fundamental para evidenciar a aplicabilidade e a importância das atividades de testes nas empresas de desenvolvimento de software participantes do estudo.

No decorrer do trabalho foram encontradas algumas dificuldades, como o alcance das empresas, limitando a pesquisa à participação de apenas cinco empresas, o que propõe expandir o número de empresas participantes para a obtenção de resultados mais significativos. Outra possibilidade seria a realização de uma análise qualitativa, que possa demonstrar como as atividades de testes auxiliam no processo de garantia da qualidade de software das empresas estudadas.

REFERÊNCIAS

- ALÉSSIO, S. C.; SABADIN, N. M.; ZANCHETT, P. S. **Processos de Software**. Unisasselv., 2017. 16, 17, 18
- ANDRADE, M. **Qualidade de Software**. Rio de Janeiro, 2015. 20
- BARBIERI, S. L. **Teste de software na indústria: um estudo qualitativo**. São Paulo, 2018. 23, 25
- BRAGA, F. A. Qualidade de software: proposta de automação de testes e de um processo ágil para uma empresa de software. **Sistemas de Informação-Florianópolis**, 2019. 11
- CAMARA, B. H. P. **Uma investigação sobre o uso de critérios de teste no desenvolvimento baseado em testes para o ensino de programação**. Cornélio Procópio, 2016. 25
- CASTILHO, A. P.; BORGES, N. R. M.; PEREIRA, V. T. **Manual de metodologia científica**. Itumbiara: ILES/ULBRA, 2011. 28
- CAUZZI, E. J. **Proposta de plano de garantia da qualidade de software para o laboratório de criação e aplicação de software**. Caxias do Sul, 2015. 23, 24
- FERNANDES, R. **Engenharia de Software**. Curitiba: Fael, 2017. 15, 16, 18, 22, 23, 25, 26, 27
- GÉRGOLI, R. **Uma análise da aplicação de testes no desenvolvimento de um sistema ERP**. São Paulo, 2017. 22, 26
- NETO, A. C. D. et al. **Caracterização do estado da prática das atividades de testes em um cenário de desenvolvimento de software brasileiro**. Rio de Janeiro, 2006. 28, 29
- NOELLO, T. P. **Melhoria no processo de teste em uma empresa de desenvolvimento de software**. Porto Alegre, 2016. 11, 25
- PERUCCI, C. C.; CAMPOS, F. C. **Técnicas de qualidade aplicadas em software: um estudo bibliométrico**. Piracicaba/SP, 2016. 11, 20
- PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software, uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016. 14, 15, 16, 17, 18, 19, 21, 23, 24, 25
- PRIMÃO A. P.; RIBEIRO, P. S. K. D. L. **Estudo de caso: técnicas de teste como parte do ciclo de desenvolvimento de software**. Alegrete/RS, 2015. 23, 26

- PRODANOV, C. C.; FREITAS, E. C. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. 2.ed. Novo Hamburgo, 2013. 28
- SALOMÃO, R. G. **Análise da relevância de testes de regressão para o mercado de desenvolvimento de software do triângulo mineiro**. Uberlândia, 2016. 27
- SAMBO, L. J. **Garantia de qualidade de software**. Sede da Universidade da África. UVA, 2017. 20
- SANTOS, D. B. **Implantação de teste de software em empresa de pequeno porte: um estudo de caso**. Marília, 2016. 11
- SANTOS, L. D. V.; OLIVEIRA, C. V. S. **Introdução à garantia de qualidade de software**. São Paulo, 2017. 22
- SCATOLINO, A. R.; CAMILO, R. D. Influência da aplicação de métodos ágeis e da gestão do conhecimento na qualidade de software: uma análise multivariada. **Revista de Gestão e Projetos**, v. 10, n. 3, 2019. 11
- SILVA, E. D. P. d.; FIGUEIREDO, L. S. **Um estudo qualitativo sobre qualidade de software no sistema de educação a distância da UFGD, o Moodle**. Sede da Universidade da África. UVA, 2019. 20
- SILVA, M. L. M. d.; DANILLO, F. D. **Uma visão geral sobre automação de testes**. **Revista Científica Multidisciplinar Núcleo do Conhecimento**. 2019. <<https://www.nucleodoconhecimento.com.br/tecnologia/automacao-de-testes/>>. Acessado em 13 de Dezembro de 2021. 23, 46
- SOMMERVILLE, I. **Engenharia de Software**. 10. ed. São Paulo: Pearson Education do Brasil, 2018. 14, 15, 16, 17, 19, 20, 21, 23, 24, 25, 26
- SOUZA, N. M. et al. **Relação entre arquitetura de software e teste de software: um mapeamento sistemático**. São Carlos - SP, 2017. 24, 26
- TRIGO, C. F. F. **Impacto da realização de testes de software**. **Tese (Mestrado em sistemas e tecnologias de informação para organizações)**. Instituto Politécnico de de Viseu, Escola Superior de Tecnologia e Gestão. Viseu, 2017. 22
- VALENTE, M. T. **Engenharia de software moderna: princípios e práticas para desenvolvimento de software com produtividade**,. 2020. 14

Apêndices

A QUESTIONÁRIO APLICADO

Analise as práticas de testes de software a seguir e assinale os pontos que melhor as definem em relação a sua aplicabilidade e importância no processo de desenvolvimento.

Definição das escalas utilizadas:

APLICABILIDADE

- (0) Não Aplicável;
- (1) Não Utilizado pela organização;
- (2) Uso Não Frequente;
- (3) Uso Habitual: Usado em grande parte dos projetos de software da organização.
- (4) Uso Padrão: Usado em todos os projetos de software da organização.

IMPORTÂNCIA

- (0) Não Importante;
- (1) Pouco importante;
- (2) Valor Limitado: Poderia ser adequado utilizá-lo;
- (3) Valor Significante: Prática recomendada;
- (4) Valor Essencial: Deve ser uma prática padrão para todos.

Questões

Q1- A empresa faz a realização de testes de software?

SIM () NÃO ()

Q2 - Existência de um responsável ou uma equipe alocada para as atividades de testes.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q3 - Realização de teste de Unidade.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q4 - Realização de teste de Integração.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q5 - Realização de teste de Sistema.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q6 - Realização de teste de Aceitação.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q7 - Realização de treinamentos sobre testes.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q8 - Separação das atividades de teste do desenvolvimento.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q9 - Existência de testadores em tempo integral para a realização dos testes.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q10 - Utilização de metodologia ou processo para sistematizar e organizar os testes.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q11 - Re-execução dos testes quando o software modificado

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q12 - Documentação do plano de teste descrevendo objetivos/abordagens.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q13 - Documentação de caso de testes e resultados.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q14 - Identificação e utilização de riscos, para projetar, organizar e executar os testes de software.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q15 - Desenvolvimento de teste antes da codificação.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2() 3() 4()

Q16 - Registro do tempo gasto em testes.

APLICABILIDADE: 0 () 1 () 2() 3() 4()

IMPORTÂNCIA: 0 () 1 () 2 () 3 () 4 ()

Q17 - Registro de falhas e defeitos detectados durante os testes.

APLICABILIDADE: 0 () 1 () 2 () 3 () 4 ()

IMPORTÂNCIA: 0 () 1 () 2 () 3 () 4 ()

Q18 - Conservação dos testes para usos futuros.

APLICABILIDADE: 0 () 1 () 2 () 3 () 4 ()

IMPORTÂNCIA: 0 () 1 () 2 () 3 () 4 ()

Q19 - Análise de defeitos identificados durante testes para descobrir suas causas e como foram introduzidos.

APLICABILIDADE: 0 () 1 () 2 () 3 () 4 ()

IMPORTÂNCIA: 0 () 1 () 2 () 3 () 4 ()

Q20 - Utilização de ferramentas geradoras de procedimentos/casos de testes.

APLICABILIDADE: 0 () 1 () 2 () 3 () 4 ()

IMPORTÂNCIA: 0 () 1 () 2 () 3 () 4 ()

Q21 - Utilização de ferramentas de gerência de teste para rastrear e registrar os resultados da execução de teste.

APLICABILIDADE: 0 () 1 () 2 () 3 () 4 ()

IMPORTÂNCIA: 0 () 1 () 2 () 3 () 4 ()

Q22 - Utilização de ferramentas de gerência de teste para acompanhar as atividades do processo de teste e os artefatos produzidos.

APLICABILIDADE: 0 () 1 () 2 () 3 () 4 ()

IMPORTÂNCIA : 0 () 1 () 2 () 3 () 4 ()