

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**QUALIDADE DE PRODUTO DE
SOFTWARE: UM ESTUDO
BASEADO EM MÉTRICAS E
AUTOMAÇÃO DE TESTES**

ROGER EMANUEL FRAGA GOMES

Trabalho de Conclusão II apresentado
como requisito parcial à obtenção do
grau de Bacharel em Sistemas de
Informação na Pontifícia Universidade
Católica do Rio Grande do Sul.

Orientador: Prof. Dr^a Sabrina Dos Santos Marczak

**Porto Alegre
2024**

DEDICATÓRIA

Dedico este trabalho à amorosa memória da minha mãe, cujo apoio e encorajamento foram a luz que me motivou e guiou os meus estudos. A você, mãe, que sempre acreditou no meu potencial e nos meus sonhos mais do que eu mesmo, dedico este pequeno grande feito. Agradeço também ao meu pai e irmã o suporte mesmo que a distância, aos amigos próximos que caminharam ao meu lado e aos professores que generosamente compartilharam seu conhecimento. Cada um de vocês teve um papel fundamental nesta jornada acadêmica. Por fim, agradeço a todos que, de alguma forma, contribuíram para que este trabalho se tornasse realidade. Que cada linha escrita represente não apenas um esforço pessoal, mas também um tributo ao amor e à valorização da educação que recebi ao longo deste caminho. Pois sou um exemplo da máxima de que a educação transforma vidas.

“Não se gerencia o que não se mede, não se mede o que não se define, não se define o que não se entende, e não há sucesso no que não se gerencia.”

(William Edwards Deming)

QUALIDADE DE PRODUTO DE SOFTWARE: UM ESTUDO BASEADO EM MÉTRICAS E AUTOMAÇÃO DE TESTES

RESUMO

Investigando a garantia da qualidade de software através de métricas específicas, foi identificado como estas contribuem para a excelência dos produtos em diferentes níveis de automação de testes. Com base na questão norteadora "Quais métricas são consideradas mais importantes para garantir a qualidade de um produto?", o estudo utilizou uma abordagem exploratória, baseada em entrevistas semi-estruturadas com profissionais da área de Tecnologia da Informação. Os dados foram analisados por meio do método de Análise Temática, revelando padrões e práticas relevantes no uso de métricas e automação para melhoria da qualidade de software. Os resultados oferecem sugestões de diretrizes práticas para organizações otimizarem suas estratégias de qualidade, promovendo maior eficiência e satisfação dos usuários finais. Este trabalho contribui para o campo ao propor um conjunto adaptável de métricas, ajustáveis ao contexto organizacional e às mudanças tecnológicas.

Palavras-Chave: qualidade de software, métricas de qualidade, desenvolvimento de produtos, automação de testes.

SOFTWARE PRODUCT QUALITY: A STUDY BASED ON METRICS AND TEST AUTOMATION

ABSTRACT

Investigating the assurance of software quality through specific metrics, it was identified how these contribute to product excellence at different levels of test automation. Based on the guiding question "Which metrics are considered most important to ensure the quality of a product?", the study used an exploratory approach based on semi-structured interviews with professionals in the Information Technology area. The data were analyzed using the Thematic Analysis method, revealing relevant patterns and practices in the use of metrics and automation to improve software quality. The results offer suggestions for practical guidelines for organizations to optimize their quality strategies, promoting greater efficiency and end-user satisfaction. This work contributes to the field by proposing an adaptable set of metrics, adjustable to the organizational context and technological changes.

Keywords: software quality, quality metrics, product development, test automation.

LISTA DE FIGURAS

Figura 2.1 – Processo de Mensuração de Produto (Adaptado de Sommerville [11] p. 675)	19
Figura 5.1 – Processo do Estudo de Campo Baseado em Entrevistas	34
Figura 5.2 – Planilha de Controle do Snowbaling	37
Figura 5.3 – Mapa Snowballing	37
Figura 5.4 – Planilha de Códigos - Análise Temática	39
Figura 5.5 – Síntese das Perspectivas sobre Qualidade - segundo a indústria de TI	50
Figura 5.6 – Síntese das Perspectivas sobre Métricas - segundo a indústria de TI	52
Figura 5.7 – Síntese das Perspectivas sobre Automação de Testes - segundo a indústria de TI	54
Figura 5.8 – Principais / Mais Importantes Métricas de Qualidade de Produto . . .	56
Figura 5.9 – Aspectos Técnicos das Métricas Identificadas	57
Figura 5.10 – Aspectos de Experiência do Usuário das Métricas Identificadas	58
Figura 5.11 – Desafios na Aplicação de Métricas	59
Figura 5.12 – Ferramentas de Execução ou de Automação de Testes	68
Figura 5.13 – Metodologias de Desenvolvimento	70
Figura C.1 – Debriefing Questions	82

LISTA DE TABELAS

Tabela 4.1 – Etapas da Análise Temática	30
Tabela 5.1 – Características dos Entrevistados	36
Tabela B.1 – Roteiro da Entrevista - Seção 1	80
Tabela B.2 – Roteiro da Entrevista - Seção 2	80
Tabela B.3 – Roteiro da Entrevista - Seção 3	81
Tabela B.4 – Roteiro da Entrevista - Seção 4	81

LISTA DE SIGLAS

ISO – International Organization for Standardization (Organização Internacional de Normalização)

BUG – Termo em Inglês comumente utilizado para se referir a um erro no código desenvolvido

TI – Tecnologia da Informação

QA – Quality Assurance / Garantidor da Qualidade

SUMÁRIO

1	INTRODUÇÃO	12
2	REFERENCIAL TEÓRICO	13
2.1	QUALIDADE DE SOFTWARE	13
2.1.1	QUALIDADE DE PRODUTO DE SOFTWARE	13
2.1.2	QUALIDADE DE PROCESSO DE SOFTWARE	15
2.2	TESTES DE SOFTWARE	15
2.2.1	TESTES MANUAIS	16
2.2.2	TESTES AUTOMATIZADOS	17
2.3	MÉTRICAS DE QUALIDADE DE SOFTWARE	18
2.3.1	DIFERENTES TIPOS DE MÉTRICAS: PRODUTO E PROCESSO	18
2.3.2	IMPORTÂNCIA DE USAR MÉTRICAS DE QUALIDADE	20
3	PESQUISA	22
3.1	DA PROBLEMÁTICA	22
3.2	DO OBJETIVO	22
3.3	DA PERGUNTA	22
4	MÉTODO DE PESQUISA	23
4.1	OBJETIVO DO ESTUDO	23
4.2	COLETA DE DADOS	23
4.3	ROTEIRO DA ENTREVISTA	23
4.4	REALIZAÇÃO DAS ENTREVISTAS	26
4.5	DEFINIÇÃO DO PERFIL DOS ENTREVISTADOS:	26
4.5.1	PROFISSIONAIS DE QUALIDADE DE SOFTWARE:	26
4.5.2	LÍDERES TÉCNICOS E GERENTES DE PROJETOS:	27
4.5.3	ESPECIALISTAS EM METODOLOGIAS ÁGEIS:	27
4.5.4	CRITÉRIOS ADICIONAIS:	27
4.6	ESTRATÉGIA DE BUSCA:	27
4.6.1	AMOSTRAGEM POR SNOWBALLING:	28
4.6.2	REDES SOCIAIS PROFISSIONAIS:	28
4.6.3	COMUNIDADES E FÓRUNS ONLINE:	29

4.7	ANÁLISE DOS DADOS OBTIDOS	29
4.7.1	TRANSCRIÇÃO E ORGANIZAÇÃO:	29
4.7.2	MÉTODO DE ANÁLISE:	30
4.7.3	RELATÓRIO DOS RESULTADOS:	31
4.8	CRONOGRAMA DE REALIZAÇÃO DAS ENTREVISTAS:	31
4.8.1	DETALHAMENTO DO CRONOGRAMA DAS ENTREVISTAS:	32
5	RESULTADOS DA PESQUISA	34
5.1	REVISÃO DO ROTEIRO / PROJETO PILOTO	34
5.2	BUSCA DOS PROFISSIONAIS / REALIZAÇÃO DAS ENTREVISTAS	35
5.3	TRANSCRIÇÃO DAS GRAVAÇÕES	38
5.4	ANÁLISE TEMÁTICA DAS TRANSCRIÇÕES / CONDENSAÇÃO DOS CÓDIGOS	38
5.5	RELATO DOS RESULTADOS	39
5.5.1	DISTRIBUIÇÃO POR IDADE:	39
5.5.2	GÊNERO:	40
5.5.3	FORMAÇÃO:	40
5.5.4	CARGOS E PRINCIPAIS RESPONSABILIDADES:	41
5.5.5	TIPOS DE EMPRESA:	42
5.5.6	TIPOS DE PROJETOS/PRODUTOS QUE ESTEJAM TRABALHANDO:	43
5.5.7	EXPERIÊNCIA COM GARANTIA DA QUALIDADE DE PRODUTO:	45
5.5.8	EXPERIÊNCIA COM AUTOMAÇÃO DE TESTE:	46
5.5.9	EXPERIÊNCIA MARCANTE COM A AUTOMAÇÃO DE TESTE:	48
5.5.10	FAMILIARIDADE COM O CONCEITO DE QUALIDADE:	50
5.5.11	FAMILIARIDADE COM O CONCEITO DE MÉTRICAS:	52
5.5.12	FAMILIARIDADE COM O CONCEITO DE AUTOMAÇÃO DE TESTES:	54
5.5.13	PRINCIPAIS / MAIS IMPORTANTES MÉTRICAS:	56
5.5.14	COMO AS MÉTRICAS SÃO UTILIZADAS:	60
5.5.15	COMO AS MÉTRICAS SÃO DEFINIDAS:	61
5.5.16	PAPEL DA AUTOMAÇÃO DE TESTES NA COLETA DAS MÉTRICAS:	63
5.5.17	LIÇÕES APRENDIDAS / BENEFÍCIOS QUE A AUTOMAÇÃO TRAZ PARA A QUALIDADE DO PRODUTO:	65
5.5.18	FERRAMENTAS DE EXECUÇÃO OU AUTOMAÇÃO DE TESTES UTILIZADAS:	67
5.5.19	METODOLOGIAS DE DESENVOLVIMENTO UTILIZADAS:	70
5.5.20	EXPERIÊNCIAS DE INSUCESSO NOS TESTES:	72

5.5.21	COMENTÁRIOS ADICIONAIS SOBRE QUALIDADE, MÉTRICAS E AUTO-MAÇÃO:	74
5.6	DISCUSSÃO DOS RESULTADOS	75
5.7	LIMITAÇÕES	76
5.8	CONCLUSÕES	77
	REFERÊNCIAS	78
	APÊNDICE A – Pedido de Consentimento	79
	APÊNDICE B – Roteiro da Entrevista	80
	APÊNDICE C – Debriefing Questions	82

1. INTRODUÇÃO

A garantia de qualidade é um componente essencial no desenvolvimento de software, impactando diretamente a aceitação e competitividade dos produtos no mercado. Produtos de software que atendem a critérios de qualidade são cada vez mais demandados, especialmente em um cenário onde a eficiência e a confiabilidade são diferenciais. Entretanto, muitas organizações ainda encontram dificuldades para implementar práticas consistentes de qualidade ao longo do ciclo de vida do desenvolvimento de software.

Este trabalho teve como foco compreender o papel das métricas de qualidade de software e da automação de testes como ferramentas para melhoria do produto final. O estudo foi desenvolvido a partir da questão central: "Quais métricas são consideradas mais importantes para garantir a qualidade de um produto?". Para isso, foi conduzido um estudo de campo baseado em entrevistas com profissionais da área de Tecnologia da Informação. A coleta de dados incluiu participantes com diferentes níveis de experiência e contextos organizacionais, com o objetivo de captar percepções diversificadas sobre o tema.

Os dados obtidos foram analisados utilizando o método de Análise Temática, permitindo identificar padrões e temas relevantes. O estudo resultou em um conjunto adaptável de métricas de qualidade, considerando suas aplicações em diferentes níveis de automação de testes. Os resultados fornecem subsídios para organizações que desejam aprimorar suas práticas de qualidade, possibilitando uma abordagem prática e ajustável às especificidades de cada contexto organizacional.

Na Seção 2, relata-se brevemente o que é qualidade de software e suas variações. A Seção 3 discorre sobre a problemática enfrentada, o objetivo do TCC e a questão norteadora. Já a Seção 4 trata sobre o método de pesquisa e avaliação dos dados qualitativos a serem identificados. E por fim, a seção 5 relata os resultados obtidos.

2. REFERENCIAL TEÓRICO

2.1 Qualidade de Software

A Qualidade de Software é uma área fundamental no desenvolvimento de sistemas e sua importância ficou mais evidente a partir do momento em que os softwares ficaram cada vez mais integrados nas atividades e cotidiano das nossas vidas [9]. Envolve a garantia de que o software seja confiável, seguro, eficiente, fácil de usar e mantido ao longo do tempo. Pois problemas com a qualidade de software datam de décadas atrás e continuam a assombrar a engenharia de software até hoje [11].

2.1.1 Qualidade de Produto de Software

A Qualidade de Produto é a área que se refere às características e propriedades de um software que determinam sua capacidade de satisfazer as necessidades explícitas e implícitas dos usuários. A percepção da qualidade de produto é subjetiva e está atrelada ao sucesso do uso do produto, que segundo Pressman [9] há uma indicação de que a efetividade da qualidade está atrelada ao valor que o produto agrega:

"... uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que o produzem e para aqueles que o utilizam."

Ainda conforme Pressman [9], existem três fatores que ajudam a definir o que é Qualidade de Produto, tais como:

- **Gestão de Qualidade Efetiva**, que estabelece a infraestrutura para construir software de alta qualidade, equilibrando mecanismos administrativos e práticas de engenharia de software.
- **Produto Útil**, que atenda às expectativas dos usuários, oferecendo conteúdo, funções e recursos desejados, juntamente com confiabilidade e ausência de erros. E,
- **Agregar Valor tanto para o Fabricante quanto para o Usuário**, quando reduz a necessidade de manutenção e suporte, agiliza os processos de negócio, tendo por consequência uma maior receita, rentabilidade e disponibilidade de informações cruciais para o negócio.

E dentre os modelos de qualidade de software existentes, destaca-se a ISO/IEC 25010 [4] que é uma norma internacional que define um modelo de qualidade de software. Que oferece um conjunto de características que podem ser utilizadas para avaliar e medir

a qualidade de um produto de software, bem como subcaracterísticas que contribuem para cada uma dessas características. Ela serve como um guia para desenvolvedores, empresas e usuários finais, auxiliando na avaliação, medição e melhoria da qualidade de produtos de software e sistemas de informação. A norma aborda oito características principais:

- **Funcionalidade**, que é a capacidade do software de cumprir suas funções previstas. Suas subcaracterísticas incluem integridade funcional (abrangência das tarefas e objetivos), correção funcional (precisão na entrega de resultados corretos) e adequação funcional (eficiência na realização de tarefas específicas).
- **Confiabilidade**, que é a capacidade de um produto desempenhar suas funções de forma consistente em condições específicas. As subcaracterísticas incluem maturidade, disponibilidade, tolerância a falhas e recuperabilidade.
- **Usabilidade**, referente à eficácia, eficiência e satisfação na realização de objetivos por meio de um produto ou sistema. Suas subcaracterísticas incluem a identificação da adequação do produto para as necessidades, facilidade de aprendizado, operabilidade, proteção contra erros do usuário, design agradável da interface e acessibilidade para diferentes perfis de usuários.
- **Eficiência**, relacionado a performance do software, à economia de recursos empregados e pode ser subdividida em três aspectos principais: desempenho temporal (tempo de resposta e tempo de processamento), consumo de recursos (quantidade e tipos de recursos utilizados) e capacidade (limite máximo suportado pelo produto ou sistema).
- **Manutenibilidade**, que é a capacidade de um produto ou sistema ser facilmente alterado, atualizado e adaptado às mudanças ambientais e aos requisitos. Suas subcaracterísticas incluem modularidade, reusabilidade, analisabilidade, modificabilidade e testabilidade, permitindo mudanças sem afetar outros componentes, a utilização de recursos em múltiplos sistemas, a identificação de problemas e partes a serem modificadas, a modificação sem introdução de defeitos e a efetividade dos testes.
- **Portabilidade**, que é a capacidade de um sistema, produto ou componente ser movido de um ambiente para outro. Suas características incluem a adaptabilidade para se ajustar a novos ambientes, a facilidade de instalação e desinstalação, e a capacidade de ser substituído por outro produto semelhante.
- **Segurança**, que é a habilidade de um produto ou sistema proteger informações e dados contra vulnerabilidades. As subdivisões desse atributo incluem confidencialidade, integridade, não repúdio, rastreabilidade de uso e autenticidade, garantindo acesso autorizado, prevenção de modificações não autorizadas, prova de eventos ocorridos, vinculação de ações a usuários autorizados e comprovação de identidade de sujeitos ou recursos.

- **Compatibilidade**, referente à capacidade de troca de informações e execução de funções em um ambiente compartilhado com outros elementos. Isso se desdobra em duas subcaracterísticas: a coexistência, que permite a realização de funções eficientemente em ambientes compartilhados, e a interoperabilidade, que possibilita a troca e uso de informações entre sistemas ou componentes diferentes.

2.1.2 Qualidade de Processo de Software

A Qualidade de Processo foca na eficácia e eficiência dos processos utilizados no desenvolvimento de software e está relacionada à capacidade dos processos de software em satisfazerem as necessidades das partes envolvidas, incluindo a definição de processos padronizados, a adoção de boas práticas de engenharia de software além da realização de revisões sistemáticas.

Segundo Sommerville [11], o gerenciamento da qualidade de processo, passa por três pontos principais: Nível Organizacional, focado em estabelecer processos organizacionais e padrões que levarão a um software de alta qualidade. Nível de Projeto, envolvendo a aplicação de processos de qualidade, realizando verificações se esses processos foram seguidos e garantindo que os resultados esperados tenham sido alcançados. E ainda no Nível de Projeto, há a preocupação de estabelecer um plano de qualidade para o nível de projeto em si. Esse plano de qualidade deve definir os objetivos de qualidade para o projeto e definir quais processos e padrões devem ser utilizados.

2.2 Testes de Software

Segundo Delmaro [7], a necessidade de realizar testes durante o processo de desenvolvimento de software surge quando da principal origem dos problemas no desenvolvimento; o erro humano. Pois dependendo principalmente da habilidade, interpretação e da execução de pessoas, os erros acabam surgindo ainda que se utilize métodos e ferramentas de engenharia de software.

Testar um software, conforme Neto [6], envolve executar o programa de maneira controlada para verificar se ele se comporta conforme especificado. O principal objetivo é identificar o máximo de falhas com o mínimo de esforço, demonstrando se os resultados estão alinhados com o esperado.

Sendo assim, os testes de software se tornam muito importantes no contexto da Engenharia de Software para assegurar características como a qualidade, confiabilidade, segurança, e custo-eficiência dos produtos de software. Contribuindo bastante para o su-

cesso de projetos de engenharia de software. Que dentre as razões para a sua utilização se podemos ter:

- **Garantia de Qualidade:** Onde os testes ajudam a identificar e corrigir bugs e defeitos no software antes de seu lançamento, garantindo que o produto final atenda aos requisitos especificados e funcione conforme o esperado.
- **Confiabilidade e Estabilidade:** Pois ao verificar se todas as partes do software funcionam corretamente sob diferentes condições, os testes asseguram a confiabilidade e estabilidade do produto, reduzindo a probabilidade de falhas críticas e ambiente de produção.
- **Economia de Custos:** Porque detectar problemas nas fases iniciais do desenvolvimento é mais barato do que corrigir erros após o lançamento. Os testes contínuos ajudam a reduzir os custos associados a retrabalho e manutenção.
- **Segurança:** Ocorrendo quando os testes ajudam a identificar vulnerabilidades e falhas de segurança, protegendo o software contra ameaças externas e garantindo a integridade dos dados dos usuários.
- **Satisfação do Usuário:** Pois um software bem testado oferece uma melhor experiência ao usuário, aumentando a satisfação e a confiança dos clientes na aplicação.

Ainda que a etapa de teste de software apresente um ganho significativo, existem desafios encontrados na prática, exigindo uma abordagem metódica no planejamento e execução de testes de software, bem como o uso de ferramentas e práticas eficazes para mitigá-los. Dentre os desafios encontrados, salienta-se o nível da Complexidade do Software, Mudanças Frequentes no Escopo dos Projetos, Limitações de Tempo e de Recursos e Ambientes de Teste Instáveis.

2.2.1 Testes Manuais

De maneira geral, segundo Crespo [3] os testes funcionais visam verificar os requisitos que definem a funcionalidade completa do software, utilizando condições de entrada e resultados esperados. Essas condições são convertidas em casos de teste e documentadas em um plano de testes.

Sendo realizados por testadores humanos sem o uso de ferramentas automatizadas, o testador executa manualmente as operações no software, verificando se o comportamento do sistema corresponde ao esperado e identificando defeitos ou falhas. Focando

e verificando se as funcionalidades do software operam conforme especificado nos requisitos, eles validam as saídas do sistema com base nas entradas fornecidas, assegurando que todas as funcionalidades previstas funcionem corretamente.

2.2.2 Testes Automatizados

Conforme Caetano [2], a automação de testes funcionais é uma técnica que complementa os testes manuais, não os substituindo, mas adicionando valor ao reduzir a repetição de testes manuais e focando em abordagens complementares. Isso resulta em uma maior abrangência e profundidade dos testes.

Ainda segundo Caetano [2] automatizar testes envolve então a criação de roteiros de teste, combinando elementos de teste e desenvolvimento de software, e requer um planejamento detalhado, sendo um projeto com suas próprias características. O autor ainda enfatiza que automatizar todos os testes manuais nem sempre é a melhor prática. Quando os testes são muito complexos ou exigem interações entre sistemas, a automação pode não ser eficaz, e esses testes devem continuar a ser feitos manualmente.

Sendo assim, a automação de testes pode melhorar a eficiência, consistência, cobertura e economia de custos, contribuindo bastante para a qualidade e sucesso dos projetos de engenharia de software. Dentre as quais podem-se destacar:

- **Eficiência e Produtividade:** Porque ao automatizar os testes há uma redução do tempo necessário para executar testes repetitivos, permitindo que a equipe se concentre em outras tarefas críticas.
- **Consistência e Confiabilidade:** Pois Testes automatizados garantem que os mesmos passos sejam seguidos de maneira consistente em cada execução, minimizando erros humanos.
- **Cobertura Ampla:** Permitindo testar mais cenários em menos tempo, aumentando a cobertura de testes e a detecção de bugs.
- **Feedback Rápido:** Fornecendo resultados rápidos após alterações no código, facilitando a identificação e correção de problemas logo no início do ciclo de desenvolvimento.
- **Redução de Custos a Longo Prazo:** Ainda que a implementação inicial possa ser cara, a automação reduz os custos de teste a longo prazo, especialmente em projetos grandes e contínuos.

A automatização de testes também apresenta uma série de problemas para ser colocada em prática. Como Custo Inicial e Tempo de Implementação porque a automação

de testes requer um investimento significativo em ferramentas, treinamento e tempo para criar e manter scripts de teste. Também pela Complexidade dos Testes pois testes complexos, especialmente aqueles que envolvem interações entre sistemas, interfaces de usuário dinâmicas ou testes em dispositivos móveis, podem ser difíceis de automatizar de forma eficaz. E também pela Escolha de Ferramentas: pois selecionar as ferramentas de automação certas para o projeto específico pode ser desafiador, considerando a diversidade de opções disponíveis e suas respectivas capacidades e limitações.

2.3 Métricas de Qualidade de Software

As Métricas de Software se referem as mais diversas formas de medidas quantitativas e qualitativas utilizadas para avaliar diferentes contextos do desenvolvimento, da qualidade e do desempenho de software. São ferramentas essenciais para a gestão e melhoria dos processos de desenvolvimento de software, permitindo a obtenção de dados objetivos e mensuráveis sobre diversos aspectos do produto e do processo de desenvolvimento.

Segundo Pressman [9], as métricas de software são um elemento-chave de qualquer processo de engenharia, pois as medições podem ser aplicadas ao processo de software com a intenção de melhorá-lo continuamente. Ainda conforme o autor, as métricas podem ser usadas durante um projeto de software para ajudar nas estimativas, no controle de qualidade, na produtividade e no controle de projeto podendo usar medidas para compreender melhor os atributos dos modelos criados e para avaliar a qualidade dos produtos ou sistemas construídos.

Métricas para o desenvolvimento de software têm alguns propósitos conforme indica Nicolette [8]. Primeiramente, podem ser usadas para avaliar o progresso em direção aos objetivos de um projeto. Em segundo lugar, podem ajudar a entender se está melhorando o desempenho de entrega. Com isso em mente, as métricas podem ajudar a guiar o trabalho em progresso, orientar melhorias de processo e fornecer uma base quantitativa para avaliar a eficácia de mudanças propostas nos métodos formais de trabalho.

2.3.1 Diferentes tipos de métricas: Produto e Processo

As métricas de qualidade de software são geralmente classificadas em dois tipos principais: métricas de produto e métricas de processo, segundo Singh, Gurdev e Singh [10]. As métricas de produto são utilizadas para avaliar as características e propriedades do software em si, como funcionalidade, confiabilidade, usabilidade, eficiência e segurança. Por outro lado, as métricas de processo são focadas nos métodos, práticas e procedimentos utilizados no desenvolvimento do software. Elas medem a eficácia, eficiência e qualidade

dos processos de engenharia de software, incluindo a produtividade da equipe, o tempo de entrega, a taxa de defeitos e a conformidade com padrões e diretrizes. Segundo Sommerville [11] a medição de software envolve, por exemplo, atribuir um valor numérico ou perfil a um atributo de um componente, sistema ou processo de software. Esses valores são comparados entre si e com padrões organizacionais para avaliar a qualidade do software ou a eficácia dos processos.

Métricas de Produto

Conforme Sommerville [11] as métricas de produto são preditivas, usadas para medir atributos ou características internas de um sistema de software. Exemplos de métricas de produto incluem o tamanho do sistema, medido em linhas de código, ou o número de métodos associados a cada classe de objeto. Ainda segundo o autor as métricas de produto se dividem em duas classes: métricas dinâmicas e métricas estáticas. As métricas dinâmicas são coletadas durante a execução de um programa e podem incluir o número de relatórios de bugs ou o tempo de conclusão de uma computação. Já as métricas estáticas são coletadas a partir de representações do sistema, como o design, programa ou documentação, e podem incluir o tamanho do código e o comprimento médio dos identificadores usados. Sommerville relata um processo de mensuração de métricas de produto, conforme descrito na Figura 2.1

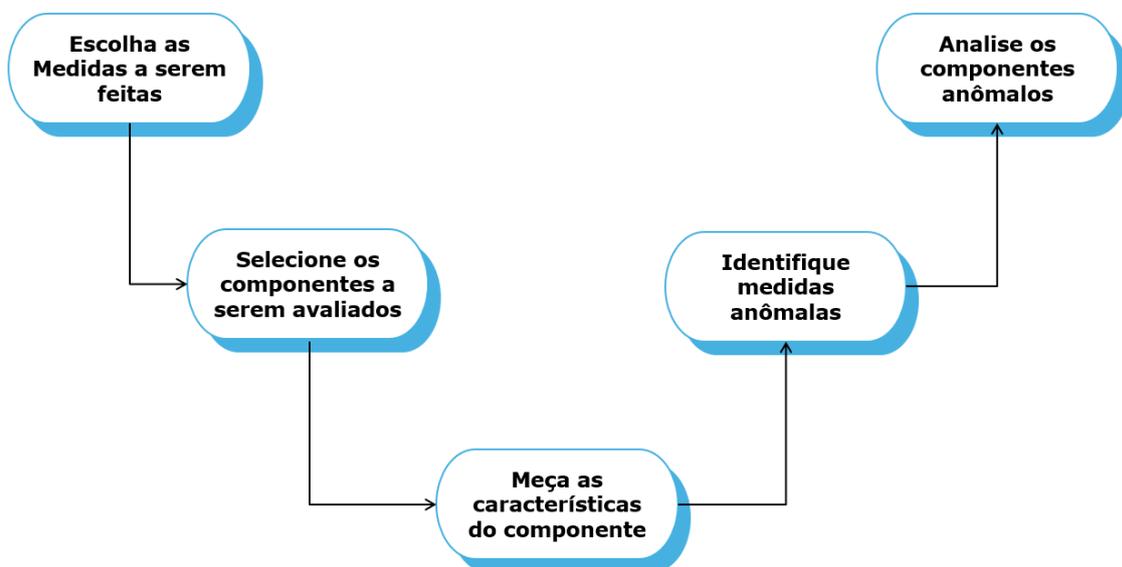


Figura 2.1 – Processo de Mensuração de Produto (Adaptado de Sommerville [11] p. 675)

Métricas de Processo

As métricas de processo são fundamentais para avaliar e melhorar o processo de software a longo prazo, segundo Pressman [9]. Elas oferecem indicadores valiosos para gerentes de projeto de software, permitindo avaliar o estado dos projetos em andamento, identificar riscos potenciais, antecipar áreas problemáticas, ajustar fluxos de trabalho e avaliar a capacidade da equipe de garantir a qualidade dos artefatos de software.

Mas embora as métricas de processo sejam essenciais, o autor reforça que é importante reconhecer que outros fatores também influenciam significativamente a qualidade do software e o desempenho organizacional, como as habilidades e motivação das pessoas envolvidas, a complexidade do produto e a tecnologia utilizada no processo de engenharia de software.

2.3.2 Importância de usar Métricas de Qualidade

O uso de métricas de qualidade de software é bastante importante para que seja possível realizar uma avaliação e aprimoramento contínuo dos processos e produtos de software. Elas permitem uma abordagem objetiva para avaliar a conformidade do software com padrões, requisitos e expectativas dos usuários. Além disso, as métricas de qualidade de software auxiliam na tomada de decisões, no planejamento estratégico e na alocação eficiente de recursos, contribuindo para a entrega de produtos de alta qualidade que atendam às necessidades e expectativas dos clientes e usuários finais.

Conforme relata Nicolette [8], um dos principais benefícios do uso de métricas de qualidade de software é poder rastrear a previsibilidade do planejamento e que isso pode melhorar a satisfação das partes interessadas. O autor ainda relata que muitas organizações enfrentam o problema de equipes de entrega que se comprometerem demais na tentativa de agradar às partes interessadas, mas depois não conseguem entregar tanto quanto prometeram. Quando as partes interessadas sabem que podem contar em receber mais ou menos o que lhes é prometido, elas se sentem confiantes na organização de entrega e oferecem maior confiança e autonomia às equipes de desenvolvimento. Nicolette ainda reforça que a experiência prática sugere uma forte correlação entre a previsibilidade do planejamento e a satisfação das partes interessadas, mesmo que não existam estudos acadêmicos formais que corroborem essa relação.

E segundo os autores Singh, Gurdev e Singh [10], as métricas de software oferecem diversas vantagens em diferentes aspectos do desenvolvimento e gerenciamento de software. Elas são úteis em estudos comparativos de metodologias de design de sistemas de software, análise e comparação de linguagens de programação, avaliação da capacidade e produtividade das pessoas envolvidas no desenvolvimento de software, elaboração

de especificações de qualidade de software e verificação da conformidade com requisitos e especificações. Além disso, as métricas ajudam a estimar o esforço necessário para o design e desenvolvimento de sistemas de software, avaliar a complexidade do código, tomar decisões sobre a divisão de módulos complexos, orientar a gestão de recursos e oferecer feedback aos gerentes de software sobre o progresso e qualidade durante o ciclo de vida do desenvolvimento do software, além de auxiliar na alocação de recursos de teste para testar o código.

Ja o contexto ágil de desenvolvimento de software possui um ponto de vista da medição de software, em que nem todas as métricas e métodos dos modelos de ciclo de vida convencionais podem ser usados sem adaptação. Pois segundo Kunz *et al.* [5], novos modelos de desenvolvimento de software precisam novas estratégias para mensuração de métricas. Portanto, o autor descreve um modelo de qualidade, métricas distintas e sua implementação em uma ferramenta de medição para gestão de qualidade no desenvolvimento ágil de software.

3. PESQUISA

3.1 Da Problemática

Assegurar a qualidade no desenvolvimento de um produto de software é fundamental, mesmo que algumas organizações não dediquem a devida atenção a essa fase do desenvolvimento. Com a crescente demanda para que as empresas invistam em produtos que ofereçam garantias de qualidade como um diferencial competitivo, um conjunto de métricas de qualidade de software torna-se a base para uma gestão eficaz da garantia de qualidade durante o desenvolvimento de produtos de software. Compreender as métricas de qualidade é vital e pode variar dependendo do contexto organizacional, como o tempo de experiência no desenvolvimento de software, o grau de automação de testes e o setor de atuação. Dentro desse escopo, além das métricas de qualidade de software, a implementação de testes automatizados também é um componente chave que será explorado para demonstrar como eles podem melhorar significativamente a qualidade do produto final.

3.2 Do Objetivo

Com base na pergunta norteadora “Quais métricas são consideradas mais importantes para garantir a qualidade de um produto?” este Trabalho de Conclusão de Curso (TCC) teve como objetivo identificar um conjunto adaptável de métricas de qualidade de produto a serem aplicadas conforme o contexto que se encontram as organizações. A fim de auxiliá-las com uma sugestão de quais métricas seriam melhor indicadas, este trabalho focou numa sugestão de direção para que essas organizações possam alcançar um bom nível de qualidade dos produtos desenvolvidos. De modo que esse conjunto de métricas, sendo adaptável, tenha sua aplicação modificada conforme os avanços ou mudanças encaradas pelas organizações. Através de um estudo de campo, baseado em entrevistas, houve o objetivo de chegar em um conjunto de métricas de qualidade de produto, servindo como material de apoio às organizações, para auxiliá-las na tomada de decisão de onde investir esforços e recursos na etapa de qualidade do produto.

3.3 Da Pergunta

Quais são as principais ou mais importantes métricas para garantir a qualidade de um produto de software?

4. MÉTODO DE PESQUISA

4.1 Objetivo do Estudo

O objetivo deste estudo de campo, baseado em entrevistas, foi identificar as métricas mais empregadas e, portanto, de grande importância para garantir a qualidade de um produto em diferentes níveis de automação de teste no processo de desenvolvimento. O estudo de campo é de natureza exploratória, utilizando uma pesquisa qualitativa conduzida por meio de entrevistas semi-estruturadas com perguntas abertas. Essas entrevistas foram realizadas com profissionais de T.I. que aplicam métricas de qualidade de software em seu trabalho diário. Os dados coletados foram analisados utilizando o método de Análise Temática, uma abordagem de análise de dados qualitativos que envolve a leitura minuciosa dos dados para identificar padrões significativos e temas relevantes. Este método requer um processo ativo de reflexividade, onde a experiência subjetiva do pesquisador desempenha um papel crucial na interpretação e compreensão dos dados.

4.2 Coleta de Dados

A coleta de dados para este estudo foi realizada por meio de um tipo de pesquisa denominado estudo de campo exploratório, baseado em entrevistas. Tais entrevistas foram individuais, semi-estruturadas, conduzidas com profissionais de Tecnologia da Informação (T.I.) que possuem experiência relevante no processos de desenvolvimento de software. Os critérios de seleção dos participantes da pesquisa considerou a experiência dos mesmos no campo do desenvolvimento de software, bem como a diversidade em relação às faixas etárias, diversidade de gênero, porte das organizações, além dos setores e áreas de atuação. O número de entrevistas planejadas foi determinado com base na saturação dos dados, sendo realizada uma entrevista piloto para validação e ajustes nos instrumentos de coleta.

4.3 Roteiro da Entrevista

O roteiro da entrevista foi elaborado com base em uma revisão da literatura sobre métricas de qualidade de software, a fim de identificar questões relevantes para obtenção de dados específicos sobre métricas em diferentes níveis de automação no desenvolvimento de software. Este roteiro foi em quatro seções distintas:

1. Introdução: Esta seção incluiu informações demográficas dos entrevistados, como idade, gênero, formação acadêmica e experiência profissional.

Ordenação	Pergunta
1.1	Qual é a sua idade?
1.2	Com qual gênero você se identifica?
1.3	Qual é a sua formação acadêmica (nível e área)?
1.4	Qual é o seu cargo atual e quais são suas principais responsabilidades?
1.5	Em que tipo de empresa você trabalha (tamanho, setor)?
1.6	Em que tipo de projetos/produtos você vem trabalhando recentemente?

2. Contextualização: Nesta etapa, os entrevistados foram questionados sobre sua experiência com o processos de desenvolvimento de software, bem como o nível de automação vivenciado, destacando suas vivências e conhecimentos na área.

Ordenação	Pergunta
2.1	Quantos anos de experiência com Garantia da Qualidade de Produto? (Quanto tu te envolvere, com que papéis e atividades esteve envolvido?)
2.2	Qual é a sua experiência com Automação de Teste?
2.3	Poderia descrever uma experiência marcante que você teve com a automação no desenvolvimento de software?
2.4	Numa escala de 0 a 10, quão familiarizado você se considera com os seguintes conceitos: de qualidade, de métricas e de automação de teste no desenvolvimento de software? Poderia explicar brevemente o que entende por cada um desses termos?
2.4.1	Quão familiarizado você se considera com o conceito de qualidade? Poderia explicar brevemente o que entende por qualidade?
2.4.2	Quão familiarizado você se considera com o conceito de métricas? Poderia explicar brevemente o que entende por métricas?
2.4.3	Quão familiarizado você se considera com o conceito de automação de testes? Poderia explicar brevemente o que entende por automação de testes?

3. Métricas de Qualidade: Na terceira etapa, foram abordadas perguntas específicas sobre as métricas consideradas mais importantes para garantir a qualidade de um produto de software. Também houve a intenção de explorar as variações dessas métricas em contextos de pouca ou muita automação.

Ordenação	Pergunta
3.1	Quais as principais (ou mais importantes) métricas que estão sendo utilizadas para garantir a qualidade de um produto de software?
3.2	Como essas métricas são utilizadas no seu ambiente de trabalho? (Ex: Como são coletadas, quem usa, para que se usa)
3.3	Como essas métricas são definidas? (Quem define? Essas definições são revisitadas de tempos em tempos?)
3.4	Como você percebe o papel da automação de testes na coleta dessas métricas? O fato de existir automação de testes pode auxiliar na coleta das métricas? (Qual o contexto? Há automação ou ela é ausente? Poderia dar exemplos?)
3.5	Quais as lições aprendidas / benefícios você acredita que a automação de testes trouxe para a sua experiência em qualidade do produto?

4. Ferramentas e Metodologias Utilizadas: Por fim, esta etapa buscou identificar as ferramentas e metodologias mais comumente associadas às métricas de qualidade de software, com o objetivo de compreender as práticas e recursos utilizados pelas organizações para garantir a qualidade de seus produtos.

Ordenação	Pergunta
4.1	Quais ferramentas de execução ou de automação de testes você utiliza no desenvolvimento de software?
4.2	Quais metodologias de desenvolvimento de software são seguidas na sua organização?
4.3	Comente sobre experiências de insucesso de testes que ocorreram na tua experiência. Situações onde os testes deram errado, não necessariamente por culpa do QA, mas ainda assim ocorreu um insucesso?
4.4	Existe algum aspecto relacionado a qualidade, métricas ou automação que não foi abordado nas perguntas anteriores e que você gostaria de comentar?

Essa estruturação do roteiro da entrevista permitiu uma abordagem abrangente e sistemática na coleta de dados, fornecendo insights importantes sobre as percepções e práticas dos profissionais de Tecnologia da Informação em relação às métricas de qualidade de software.

4.4 Realização das Entrevistas

A realização das entrevistas foi conduzida seguindo um processo estruturado, considerando duas fases distintas onde houve o planejamento das entrevistas bem como a aplicação prática das mesmas, conforme descrito a seguir:

1. Planejamento das Entrevistas: Nesta etapa, o roteiro da entrevista foi validado, para abordar de maneira eficaz os temas pertinentes à pesquisa. A validação se deu por meio de uma entrevista piloto para ajustes e refinamentos no roteiro, garantindo sua eficiência e relevância.
2. Aplicação das Entrevistas: As entrevistas foram conduzidas de forma remota, utilizando a ferramenta Google Meet. Caso fosse necessário, seriam realizadas presencialmente, conforme a disponibilidade dos participantes. Durante as entrevistas, foi reforçado o consentimento dos entrevistados, bem como foi assegurada a confidencialidade dos dados coletados. As entrevistas foram gravadas para garantir precisão na análise e foram feitas anotações complementares durante as entrevistas para capturar informações adicionais e contextualizar as respostas dos participantes.

Essa abordagem na realização das entrevistas permitiu a coleta de dados de forma organizada, ética e precisa, contribuindo significativamente para a qualidade e validade dos resultados obtidos na pesquisa.

4.5 Definição do Perfil dos Entrevistados:

Como forma de garantir que a pesquisa tenha obtido dados o mais relevantes possíveis e sobre os processos de desenvolvimento de software, métricas de qualidade e automação de testes, os entrevistados devriam possuir os seguintes perfis:

4.5.1 Profissionais de Qualidade de Software:

- QA - Quality Assurance: Profissionais responsáveis por garantir a qualidade do software através de práticas de automação, criação e execução de testes, e validação de processos de desenvolvimento.
- Analistas de Testes: Profissionais que planejam, projetam e executam testes para identificar defeitos e garantir que os produtos de software atendam aos requisitos especificados.

- Testadores: Profissionais que executam casos de teste, relatam bugs e trabalham em estreita colaboração com desenvolvedores para resolver problemas de qualidade.

4.5.2 Líderes Técnicos e Gerentes de Projetos:

- Líderes Técnicos: Profissionais responsáveis por liderar equipes de desenvolvimento e garantir a implementação de práticas de qualidade e automação.
- Gerentes de Projetos de Software: Profissionais que gerenciam projetos de desenvolvimento de software, com foco na integração de práticas de qualidade e automação nos processos de desenvolvimento.

4.5.3 Especialistas em Metodologias Ágeis:

- Scrum Masters: Profissionais que facilitam a aplicação de metodologias ágeis nas equipes de desenvolvimento, garantindo que as práticas de qualidade e automação sejam integradas nos processos de trabalho.

4.5.4 Critérios Adicionais:

- Experiência Profissional: Mínimo de 1 ano de experiência relevante na área de desenvolvimento de software e/ou qualidade de software.
- Contribuições em Projetos de Automação: Experiência comprovada em projetos que envolvam a automação de testes e processos de desenvolvimento de software.
- Conhecimento de Ferramentas e Metodologias: Familiaridade com ferramentas de automação de testes e metodologias de desenvolvimento ágil.

4.6 **Estratégia de Busca:**

Para encontrar e selecionar profissionais adequados para as entrevistas, será utilizada a Técnica de Pesquisa Snowballing. Além de pesquisas em redes sociais profissionais como LinkedIn bem como comunidades e fóruns online como GitHub e Stack Overflow. Uma triagem inicial de perfis será executada, seguida de contatos privados para verificar o interesse dos candidatos em participar da entrevista.

4.6.1 Amostragem por Snowballing:

É uma técnica de pesquisa utilizada para identificar e recrutar participantes para um estudo. Segundo Baldin [1], começando com um pequeno grupo de indivíduos já conhecidos ou facilmente acessíveis (os "semeadores") e, a partir deles, obtendo indicações de novos participantes. Este processo se repete, formando uma "bola de neve" que cresce à medida que mais pessoas são incluídas na pesquisa.

Perfil do Entrevistado:

Como Aplicar a Amostragem por Snowballing:

- Identificação dos Semeadores: O início se deu identificando alguns indivíduos que atendam ao perfil desejado. Esses foram contatos pessoais, colegas de trabalho ou conhecidos na indústria.
- Primeiro Contato: Foi necessário convidar esses semeadores para participar da pesquisa e, após a entrevista, solicitar que indicassem outros profissionais que também atendam ao perfil desejado.
- Expansão da Rede: Ao entrar em contato com as novas indicações e repetir o processo, foi solicitado a cada novo participante que indicasse mais pessoas qualificadas.
- Continuação do Processo: Foi expandida a rede de participantes até ter alcançado um número suficiente de entrevistas ou até que não houvesse mais novas indicações significativas.

O uso do Snowballing permitiu alcançar participantes que poderiam ser difíceis de encontrar através de métodos tradicionais, especialmente em nichos ou comunidades específicas. Reduziu o tempo e os esforços necessários para identificar e recrutar novos participantes, aproveitando as redes de contatos dos próprios entrevistados. Além dos participantes indicados por colegas ou conhecidos estarem mais dispostos a participar devido à confiança estabelecida nas recomendações.

4.6.2 Redes Sociais Profissionais:

LinkedIn:

Foram utilizados filtros de pesquisa avançada para encontrar profissionais com os cargos e habilidades desejadas. Grupos relacionados a qualidade de software e desenvolvimento ágil também foram consultados para encontrar candidatos.

Exemplo de mensagem inicial:

"Olá [Nome],

Estou conduzindo um Estudo de Campo baseado em Entrevistas, como TCC da minha faculdade, sobre qualidade de software, focando em métricas de qualidade e automação de testes, e gostaria de saber se você estaria interessado em participar de uma entrevista. Sua experiência como [Cargo] na [Empresa] seria extremamente valiosa para nosso estudo.

Agradeço desde já pela consideração.

Atenciosamente, Roger Gomes - Acadêmico de Sistemas de Informação / PUCRS"

4.6.3 Comunidades e Fóruns Online:

GitHub:

Identificado profissionais com o perfil desejado que contribuíram para projetos relacionados a qualidade de software e automação de testes.

Stack Overflow:

Utilizado tags relacionadas a automação de testes, QA, e qualidade de software para encontrar usuários ativos.

Reddit:

Participado de subreddits como r/QualityAssurance, r/SoftwareTesting, e r/devops para postar convites para entrevistas.

4.7 Análise dos Dados Obtidos

O processo de análise dos dados obtidos neste estudo iniciou-se com a transcrição das entrevistas utilizando uma ferramenta de inteligência artificial para apoio, em seguida, os dados são codificados para organização e posterior análise.

4.7.1 Transcrição e Organização:

O processo de transcrição e organização dos dados foi conduzido de forma sistemática e muito cuidadosa, seguindo etapas como Transcrição e Codificação, as quais são detalhadas abaixo:

- As entrevistas foram transcritas utilizando uma ferramenta de inteligência artificial (VEED.io), que auxiliou na obtenção dos dados brutos, que passaram por uma revisão fina de correções e uma revisão final acompanhada da gravação da entrevista.
- Os dados dos participantes foram codificados conforme o Método de Análise Textual Temática além de garantir a confidencialidade das informações coletadas durante as entrevistas. Essas etapas foram bastante importantes para assegurar a integridade e a privacidade dos dados, bem como para facilitar a análise posterior dos dados dentro do contexto da pesquisa científica.

4.7.2 Método de Análise:

Foi utilizado o método de Análise Textual Temática para a análise dos dados. Esse método foi previsto para ser realizado em seis etapas, as quais estão relacionadas na Tabela 4.1.

Nome da Etapa	Descrição da Etapa
Familiarização com os dados	As transcrições foram revisadas diversas vezes
Geração de códigos iniciais do objeto	Frases ou pedaços interessantes de frases foram destacados e atribuídos códigos a eles
Busca por temas	Foram códigos que apareceram com bastante frequência e foi possível de serem categorizados em temas mais generalistas
Revisão de temas	Houve a revisão dos temas e foi decidido se eles seriam descartados ou mantidos
Definição e nomeação de temas	Nomes foram atribuídos aos temas para auto explicar o seu conteúdo / significado
Produção do relatório	A análise foi formalizada por meio de uma planilha de controle contendo os principais achados e citações que sustentavam cada tema

Tabela 4.1 – Etapas da Análise Temática

A análise seguiu um processo iterativo: as etapas 2 e 5 foram realizadas diversas vezes até que a temática fosse saturada.

- Acredita-se que os dados apresentam um bom nível de saturação, uma vez que foram identificados repetições de vários temas relatados nas entrevistas.

4.7.3 Relatório dos Resultados:

O relatório dos resultados desta pesquisa foi estruturado na seção 5, visando uma análise robusta dos dados obtidos, contemplando as etapas decorridas do processo como um todo o estudo com especial destaque para o relato dos resultados propriamente ditos. Seguindo a seguinte ordem:

1. Revisão do Roteiro / Execução do Projeto Piloto.
2. Busca dos Profissionais / Realização das Entrevistas.
3. Transcrição das Entrevistas.
4. Análise Temática das Transcrições / Condensação dos Códigos.
5. Relato dos Resultados.
6. Discussão dos Resultados.
7. Limitações Enfrentadas.
8. Conclusões

4.8 Cronograma de Realização das Entrevistas:

O seguinte cronograma era a ideia prevista para ser seguido ao longo da execução do Estudo de Campo. Para que entrevistas fossem realizadas de forma organizada, com tempo reservado para a transcrição e análise dos dados, sem ocupar o semestre inteiro apenas com as entrevistas. Além disso, a estrutura semanal ajudaria a manter um ritmo constante e a evitar acúmulo de tarefas. Na prática houve a tentativa de segui-lo, mas devido a problemas pessoais ao longo do semestre o cronograma serviu de referencial não sendo seguido totalmente.

De forma resumida o cronograma contaria com as seguintes etapas: Em Agosto: Preparação e início das entrevistas; Em Setembro: Continuação e finalização das entrevistas; Outubro: Transcrição e Análise dos Dados; Novembro: Conclusão da análise e Redação do Relatório Final.

4.8.1 Detalhamento do Cronograma das Entrevistas:

Julho:

Semana 3-4 (15 a 31 de julho): Projeto Piloto

- Realização de entrevistas teste para validação final do roteiro da entrevista.
- Contatos iniciais para identificação de possíveis semeadores.

Agosto:

Semana 2 (12 a 18 de agosto): Preparação

- Revisão Final do roteiro de entrevista após Projeto Piloto.
- Continuação da identificação dos semeadores e dos contatos preliminares.
- Agendamento das primeiras entrevistas.

Semana 3-4 (19 a 31 de agosto): Primeiras Entrevistas

- Realização das primeiras entrevistas (2 entrevistas por semana no mínimo).
- Transcrição imediata das entrevistas realizadas.
- Solicitação de indicações para a estratégia de snowballing.

Setembro:

Semana 1-2 (2 a 15 de setembro): Continuação das Entrevistas

- Realização de entrevistas (2 entrevistas por semana no mínimo).
- Transcrição imediata das entrevistas realizadas.
- Análise preliminar dos dados para identificar temas emergentes.

Semana 3-4 (16 a 30 de setembro): Expansão e Conclusão das Entrevistas

- Continuação e finalização das entrevistas restantes (2 entrevistas por semana no mínimo).
- Transcrição imediata das entrevistas realizadas.
- Solicitação de últimas indicações se necessário.

Outubro:

Semana 1-2 (1 a 13 de outubro): Finalização das Entrevistas e Início da Análise

- Conclusão das últimas entrevistas, se houver alguma pendente.
- Transcrição das últimas entrevistas.
- Revisão e organização de todas as transcrições.
- Início da análise detalhada dos dados coletados.

Semana 3-4 (14 a 31 de outubro): Análise dos Dados

- Continuação da análise dos dados, categorização e identificação de padrões.
- Elaboração de relatórios preliminares sobre os resultados encontrados.

Novembro:

Semana 1-2 (4 a 16 de novembro): Finalização da Análise e Redação

- Conclusão da análise dos dados.
- Redação do relatório final baseado nos resultados.
- Revisão e edição do relatório/artigo.

Semana 3 (17 a 24 de novembro): Conclusão do Projeto

- Finalização da redação e preparação para a apresentação da Banca.
- Preparação de materiais complementares (slides, gráficos, etc.).
- Apresentação dos resultados na Banca.

5. RESULTADOS DA PESQUISA

São apresentados nesta seção as etapas, conforme a Figura 5.1, das metodologias utilizadas que foram fundamentais para a realização do estudo de campo, detalhando o processo de desenvolvimento e aplicação da pesquisa. Inicialmente, o roteiro da pesquisa passou por uma revisão criteriosa, seguida da realização de um projeto piloto para validar sua clareza e funcionalidade. A partir daí, deu-se início à busca pelos profissionais, realizada com o método de snowballing, e à condução das entrevistas, que foram realizadas em ambiente virtual com o uso de estratégias para minimizar imprevistos técnicos.

As entrevistas foram transcritas utilizando ferramentas de inteligência artificial, com ajustes manuais para assegurar a precisão dos registros. Posteriormente, essas transcrições passaram por uma análise temática detalhada, que permitiu identificar os principais temas e consolidar os dados em códigos bem definidos. Por fim, os resultados obtidos ao longo de todo o processo foram organizados e interpretados, fornecendo uma base sólida para as conclusões do estudo.



Figura 5.1 – Processo do Estudo de Campo Baseado em Entrevistas

5.1 Revisão do Roteiro / Projeto Piloto

O roteiro inicial da pesquisa passou por uma revisão detalhada, durante a qual foram realizados ajustes para garantir clareza, objetividade e alinhamento com os objetivos do estudo. Após essa etapa, foi conduzido um projeto piloto para validar o roteiro em prática.

O projeto piloto consistiu na realização de uma entrevista com um parceiro do estudo que, embora conhecesse a ideia geral da pesquisa, não tinha familiaridade com o roteiro em si. Essa entrevista proporcionou a oportunidade de colocar o roteiro em prática e avaliar tanto a clareza das perguntas quanto a condução da entrevista.

Essa experiência foi fundamental para identificar pontos de melhoria. Foram ajustados aspectos como a redução de perguntas redundantes, a reformulação de algumas questões e a criação de novas perguntas que surgiram como necessárias durante o piloto. Esses refinamentos resultaram na versão final do roteiro, utilizada em todas as entrevistas subsequentes.

Ao longo das entrevistas reais, observou-se que cada pergunta representava uma oportunidade única para obter informações. Refazer uma pergunta ou tentar explicá-la novamente poderia transmitir a ideia de que o entrevistado não havia compreendido, o que não era desejável. Assim, o piloto também ajudou a aprimorar a habilidade de conduzir a entrevista de forma clara e assertiva, respeitando a dinâmica natural do diálogo.

5.2 Busca dos Profissionais / Realização das Entrevistas

Para a seleção dos participantes da pesquisa, utilizou-se o método de snowballing. Foram contatados profissionais que atendiam ao perfil previamente definidos para o estudo. Esses profissionais, por sua vez, acabaram indicando outras pessoas qualificadas, ampliando a rede de participantes de forma progressiva. Esse método se mostrou eficiente para alcançar uma diversidade de perspectivas e garantir uma amostra significativa.

Ao contatar os profissionais foram utilizados textos de acordo com a ocasião:

Texto de Anúncio Geral

"Olá pessoal! Me chamo Roger e estou finalizando a minha (tão sonhada) graduação em Sistemas de Informação e meu TCC busca informações sobre Métricas de Qualidade de Produto. Estou realizando entrevistas que vão me ajudar imensamente nesse Estudo de Campo e procuro pessoas para participarem da minha pesquisa. Meu público alvo são Pessoas Analistas de Testes / QAs / Testadores / Gestores desses cargos. Peço e gostaria de agradecer qualquer indicação de pessoas para que eu possa entrar em contato e conversar sobre a entrevista. Fiquem bem e obrigado!"

Texto de Conexão LinkedIn

"Olá nomePessoa! Peguei seu contato com o(a) nomeIntermediário. Somos colegas de trabalho (Indicar vínculo). Notei que teu perfil é IDEAL para uma Pesquisa Científica que eu estou conduzindo. Posso te fazer um convite?"

Texto de Convite Direto

"Olá nomePessoa! Meu nome é Roger e sou colega/amigo do da nomeIntermediário. Estou finalizando a minha graduação em Sistemas de Informação e meu TCC busca informações sobre Métricas de Qualidade de Produto. Estou realizando entrevistas que vão me ajudar imensamente nesse Estudo de Campo e tu tens o perfil IDEAL que eu procuro! Gostaria de saber se tu aceitarias participar dessa entrevista pro meu TCC sobre Métricas de Qualidade! Não é uma entrevista longa, cerca de 30 a 40 minutos e eu conseguirei te perguntar sobre todos os pontos que eu preciso. Tenho certeza que tu serias de grande valia para a minha pesquisa. Por isso faço votos que tu participe! aguardo um retorno seu, inclusive se tiver alguma dúvida estou a disposição. Obrigado!"

Os participantes da pesquisa apresentaram um perfil profissional diversificado, com formações acadêmicas em áreas como Ciência da Computação, Engenharia de Software e Sistemas de Informação. A maioria dos entrevistados atuava como desenvolvedores, analistas de testes e engenheiros de qualidade. O tempo de experiência na área de qualidade também variou entre os participantes, com alguns acumulando mais de 10 anos de atuação e outros com menos de 5 anos, demonstrando a amplitude de perfis profissionais que contribuíram para a pesquisa. Conforme a Tabela 5.1 identificando os entrevistados pelo número da sua entrevista:

Entrevista	Código	Cargo	Anos Área de Qualidade
01	In3-PA-FP	Analista de Testes	3 anos
02	In8-JA-SR	Engenheira de Qualidade Sênior	6 anos
03	In17-SM-CD	Quality Engineer 1	3 anos
04	In12-AS-GS	Analista de Qualidade	3 anos
05	In14-JS-PF	Engenheiro de Automação	5 anos e três meses
06	In13-AL-PO	Analista de Testes Sênior	13 anos
07	In18-GS-GJ	Technical Product Manager	1 ano e meio
08	In19-PC-SC	Product Manager	4 anos
09	In22-PA-CG	QA Engineer Júnior	3 anos
10	In20-PC-DC	Líder de Projetos	7 anos
11	In25-JS-SG	Analista de Testes Sênior	10 anos
12	In10-GP-GL	Analista de Testes Júnior	2 anos e meio
13	In27-EA-CS	Analista de Testes Sênior	15 anos

Tabela 5.1 – Características dos Entrevistados

Para organizar e controlar esse processo, foi desenvolvida uma planilha detalhada conforme a Figura 5.2, onde foram registrados os dados relevantes dos entrevistados. Essa planilha incluía informações como o nome do participante, o responsável pela indicação, e o status de cada etapa do estudo (entrevista realizada, transcrição concluída, análise em andamento). A partir dessa base, foi criado um mapa visual, Figura 5.3, que condensava as informações de forma gráfica, facilitando a análise e o acompanhamento geral do progresso da pesquisa.

Snowballing - Estudo de Campo

Interação			Intermediário		Indicado		Status		
Código	Data	Intervist.	Nome	Contato	Nome	Contato	Indicação	Entrevista	Transcrição
In1-EP-N	13/09/2024						Não Respondeu	-	-
In2-JS-S	13/09/2024						Indicou	-	-
In3-PA-FP	13/09/2024	1					Aceitou	Realizada	Realizada
In4-JA-S	13/09/2024						Indicou	-	-
In5-JA-AZ	14/09/2024						Não Respondeu	-	-
In6-JA-FV	14/09/2024						Recusou	-	-
In7-JA-AR	14/09/2024						Recusou	-	-
In8-JA-SR	14/09/2024	2					Aceitou	Realizada	Realizada
In9-JA-GT	14/09/2024						Não Respondeu	-	-
In10-GP-GL	16/09/2024	12					Aceitou	Realizada	Realizada
In11-AS-AS	17/09/2024						Não Respondeu	-	-
In12-AS-GS	17/09/2024	4					Aceitou	Realizada	Realizada
In13-AL-PO	17/09/2024	6					Aceitou	Realizada	Realizada
In14-JS-PF	19/09/2024	5					Aceitou	Realizada	Realizada
In15-PA-DC	20/09/2024						Não Respondeu	-	-
In16-PA-LB	20/09/2024						Não Respondeu	-	-
In17-SM-CD	20/09/2024	3					Aceitou	Realizada	Realizada
In18-GS-GJ	24/09/2024	7					Aceitou	Realizada	Realizada
In19-PC-SC	25/09/2024	8					Aceitou	Realizada	Realizada
In20-PC-DC	25/09/2024	10					Aceitou	Realizada	Realizada
In21-WC-N	30/09/2024						Sem Indicações	-	-
In22-PA-CG	02/10/2024	9					Aceitou	Realizada	Realizada
In23-PA-KT	02/10/2024						Não Respondeu	-	-
In24-JS-MA	03/10/2024						Não Respondeu	-	-
In25-JS-SG	03/10/2024	11					Aceitou	Realizada	Realizada
In26-CG-LB	08/10/2024						Não Respondeu	-	-
In27-EA-CS	07/11/2024	13					Aceitou	Realizada	Realizada

Figura 5.2 – Planilha de Controle do Snowbaling

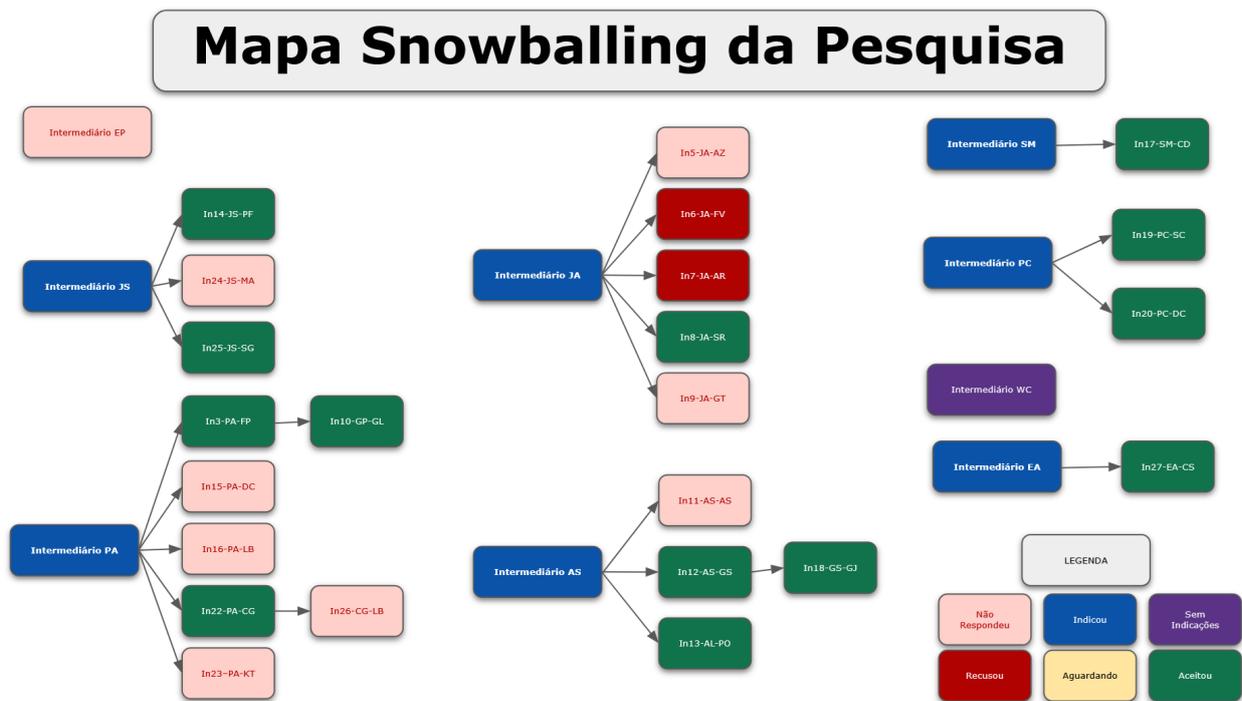


Figura 5.3 – Mapa Snowballing

As entrevistas foram realizadas por meio da plataforma Google Meet. Para garantir a pontualidade e minimizar imprevistos técnicos, a sala virtual sempre foi acessada com antecedência e foi preparado um link de internet secundário como medida de contingência. Durante as entrevistas, foi seguido o roteiro porém procurando oportunidades de flexibilidade para explorar respostas mais aprofundadas e detalhes relevantes, conforme a necessidade. As perguntas abertas foram fundamentais para incentivar os participantes a compartilharem suas experiências e percepções de maneira ampla, enriquecendo os dados coletados. Um pedido de consentimento também foi realizado no início de cada entrevista,

para obter a permissão de gravação da entrevista, cuja finalidade do uso dos dados foi expressa como sendo apenas para uso acadêmico bem como as informações pessoais e imagem dos entrevistados seriam mantidas sob sigilo.

5.3 Transcrição das Gravações

As gravações das entrevistas foram transcritas utilizando a ferramenta de inteligência artificial VEED (www.veed.io), que auxiliou na geração inicial dos textos a partir do áudio. Essa ferramenta se mostrou eficiente para obter uma versão preliminar das falas, acelerando o processo de transcrição.

No entanto, após a obtenção dos textos brutos, foi necessário realizar diversos ajustes. Cada falante foi devidamente identificado, e foram incluídos marcadores para distinguir as participações durante o diálogo. Além disso, os parágrafos foram tabulados para melhorar a organização textual, e pequenos ajustes gramaticais foram aplicados, uma vez que a transcrição automatizada não era totalmente precisa.

Após esses ajustes iniciais, uma revisão detalhada foi conduzida. Essa etapa consistiu em acompanhar o texto transcrito enquanto ouvia novamente a gravação, realizando uma comparação palavra por palavra, em um processo minucioso semelhante a um "pente fino". Essa revisão final garantiu que todas as informações fossem fidedignas ao áudio original e que eventuais erros remanescentes fossem corrigidos.

Durante a transcrição, foi surpreendente perceber a quantidade de vícios de linguagem presentes nas conversas, em ambas as falas (entrevistador e entrevistados). Essa constatação trouxe reflexões importantes sobre os padrões de comunicação utilizados durante o estudo.

5.4 Análise Temática das Transcrições / Condensação dos códigos

A análise das entrevistas seguiu uma abordagem textual temática conforme previsto, permitindo identificar os principais temas emergentes nas respostas dos participantes. Cada transcrição foi examinada individualmente, com uma leitura detalhada de cada pergunta e respectiva resposta. Esse processo visou extrair os elementos centrais de cada discurso, alinhando-os aos objetivos do estudo.

Os temas principais foram destacados diretamente no arquivo de texto correspondente, utilizando comentários e marcadores para facilitar a organização. Essa anotação temática proporcionou uma visão clara das ideias mais recorrentes e significativas, bem como das variações nas perspectivas apresentadas pelos entrevistados.

Para garantir uma organização ainda mais eficiente, os códigos temáticos foram condensados em uma planilha de controle, de acordo com a Figura 5.4. Essa planilha foi estruturada para agrupar e revisar os códigos de forma sistemática, permitindo uma análise mais objetiva e coesa. Durante essa revisão final, códigos maiores foram refinados para melhor representar os dados.

O resultado final foi uma síntese das informações mais significativas, estruturadas de maneira a sustentar as conclusões da pesquisa e a facilitar a interpretação dos dados.

Perguntas	1	Entrevista 2	Entrevista 3	Entrevista 4
		Código	Código	Código
1.1 Idade	32	32	23	23
1.2 Gênero que se identifica		- Feminino	- Feminino	- Masculino
1.3 Formação Acadêmica / Nível e Área	ago em ADS	- Bacharelado em Engenharia de Software; - Pós-graduação em Business Intelligence;	- Mestrado, em andamento, em Ciência da Computação;	- Engenharia de Software;
1.4 Cargo Atual		- Engenheira de Qualidade Sênior;	Quality Engineer 1	- Analista de Qualidade
1.4 Principais Responsabilidades	de; dução;	- Análise de demandas focada em qualidade; - Elaboração de cenários e requisitos; - Desenvolvimento de automação de testes; - Melhoria de processos e cultura de qualidade;	- Testes manuais; - Testes automatizados (front e back-end); - Testes de banco de dados;	- Testes front-end mobile; - Testes automatizados; - Planejamento dos testes executados;
Complementar (Entrevista 6) - Estimativa do quanto trabalha com testes manuais e automatizados				
1.5 Tipo de empresa / tamanho e setor	terceiro setor			- Grande porte; - Multinacional; - Setor de bebidas;
1.6 Projetos que trabalha atualmente	para Solicitação ros em produção; reções;	- Projeto de Integração de Processos (iPaas); - Conectores e Tecnologias; - Complexidade do Projeto;	- Projetos da área financeira; - Projetos que envolvem mobile, android e iOS; - Projetos que envolvem plataforma de salesforce / portal de aplicações;	- Digitalização e automação de processos de vendas; - Apoio à autonomia dos representante de vendas; - Eficiência operacional;
Complementar (Entrevista 2) - Nível de complexidade das integrações;		- Complexidade de Integração; - Entendimento Abrangente; - Seleção de Cenários Prioritários;		

Figura 5.4 – Planilha de Códigos - Análise Temática

5.5 Relato dos Resultados

5.5.1 Distribuição por idade:

A faixa etária dos entrevistados variou, mostrando uma diversidade de experiências e estágios de carreira entre os profissionais de tecnologia entrevistados. A idade média dos participantes foi de aproximadamente 30 anos, com uma variação de 23 a 42 anos.

Os dados revelaram uma predominância de profissionais na faixa dos 30 a 33 anos, representando uma parcela significativa dos entrevistados. Observou-se ainda a presença de alguns profissionais mais jovens, na casa dos 23 a 25 anos, e outros na faixa de 36 a 42 anos, o que reflete a diversidade na experiência profissional e, possivelmente, na perspectiva sobre métricas de qualidade e automação em testes.

5.5.2 Gênero:

A amostra contou com participantes que se identificaram principalmente como masculino e feminino, refletindo a diversidade dos profissionais na área de tecnologia. Entre os entrevistados, a maioria (aproximadamente 60%) se identificou como masculino, enquanto cerca de 40% se identificaram como feminino.

Essa distribuição indica uma representatividade de diferentes perspectivas de gênero, permitindo uma visão mais inclusiva e ampla sobre as métricas de qualidade de produto e a automação de testes.

5.5.3 Formação:

A formação acadêmica dos profissionais entrevistados varia, abrangendo diferentes níveis e áreas, o que reflete uma diversidade de perfis e qualificações na área de tecnologia.

Entre os participantes, os principais níveis de formação encontrados foram:

- **Bacharelado:** Grande parte dos entrevistados [4, 5, 6, 12 e 13] possui bacharelado em áreas diretamente relacionadas à tecnologia, como Engenharia de Software e Sistemas de Informação. Esse grupo representa profissionais com uma sólida base técnica e uma visão abrangente do desenvolvimento de software.
- **Pós-graduação:** Alguns participantes [2, 8 e 10] destacaram formações em nível de pós-graduação, principalmente em áreas como Business Intelligence, Liderança e Management 4.0, e Engenharia de Software. Esse nível de qualificação indica uma busca por especialização e aprofundamento em temas específicos de gestão e tecnologia.
- **Mestrado em andamento:** Houve também participantes [3 e 7] que estão cursando ou concluindo um mestrado, especialmente em Ciência da Computação, com foco em sub-áreas como Engenharia de Requisitos. Esses profissionais estão em busca de conhecimentos avançados e especializações mais profundas dentro do campo da tecnologia.
- **Outras formações:** Também foram identificadas formações variadas [1, 9 e 11], como técnicos em desenvolvimento de software e tecnólogos em áreas específicas, como Sistemas para Internet e Análise de Sistemas. Em casos específicos, houve profissionais que iniciaram a carreira em áreas distintas, como Educação Física, e migraram para a tecnologia por meio de cursos técnicos e especializações.

Os resultados mostram uma variedade de trajetórias acadêmicas, indicando que o campo de métricas de qualidade e testes automatizados atrai profissionais com diversas bases educacionais e níveis de especialização, o que enriquece o contexto e a abordagem dos temas investigados.

5.5.4 Cargos e Principais Responsabilidades:

Os participantes ocupam cargos variados no setor de tecnologia, relacionados ao controle de qualidade, automação de testes e gestão de produtos. Abaixo, as funções foram organizadas em categorias, com um resumo das principais responsabilidades:

- Analistas de Testes e Qualidade [1, 4, 6, 11, 12 e 13]
 - Cargos:
 - * Analista de Testes, Analista de Qualidade, Analista de Testes Sênior
 - Principais Responsabilidades:
 - * Realização de testes manuais e automatizados, incluindo testes funcionais, regressivos e de front-end mobile.
 - * Planejamento de cenários de teste e execução de testes para prevenir defeitos em produção.
 - * Organização e melhoria dos processos de qualidade, com foco em orientar analistas juniores e definir ferramentas adequadas.

- Engenheiros de Qualidade e Automação [2, 3, 5 e 9]
 - Cargos:
 - * Engenheira de Qualidade Sênior, Quality Engineer 1, QA Engineer Júnior, Engenheiro de Automação
 - Principais Responsabilidades:
 - * Desenvolvimento de automação de testes (front-end e back-end), incluindo testes de banco de dados.
 - * Análise de demandas focadas em qualidade, criação de cenários e requisitos de teste, e desenvolvimento de processos para fortalecer a cultura de qualidade.
 - * Gestão do ecossistema de automação e suporte direto aos desenvolvedores, além de validação de requisitos e execução de testes funcionais.

- Gerentes de Produto e Projetos [7, 8 e 10]
 - Cargos:
 - * Technical Product Manager, Product Manager, Líder de Projetos
 - Principais Responsabilidades:
 - * Gestão de produtos, incluindo a administração de mudanças e ajustes em produtos já no mercado.
 - * Coordenação de projetos em andamento e desenvolvimento de estratégias de management 3.0 para promover pensamento inovador e cuidado com as equipes.
 - * Supervisão do ciclo de vida dos produtos, desde a implementação de ajustes até a melhoria contínua do produto.

5.5.5 Tipos de Empresa:

Os profissionais entrevistados atuam em empresas de variados tamanhos, setores e modelos de negócio, refletindo a diversidade do mercado de tecnologia no Brasil. As empresas foram classificadas em termos de porte, setor de atuação e especificidades de mercado, conforme descrito a seguir:

- Porte das Empresas: As organizações representadas na pesquisa incluem tanto empresas de médio porte quanto de grande porte, com uma leve predominância destas últimas.
 - Médio porte [1, 5, 8, 10 e 12]: Identificadas em setores como educação, supply chain, crédito estudantil e empresas em expansão.
 - Grande porte [3, 4, 7, 9, 11 e 13]: Encontradas em segmentos como bebidas, agroindústria, setor bancário, varejo e atacado, além de empresas multinacionais.
- Setores de Atuação: A análise revelou uma ampla gama de setores em que os entrevistados atuam, demonstrando a aplicação transversal de práticas de qualidade e automação no desenvolvimento de software. Entre os setores identificados, destacam-se.
 - Tecnologia para o terceiro setor [1, 5 e 12]: Fundação sem fins lucrativos.
 - Agroindústria [9]: Empresas que operam em atividades relacionadas ao agronegócio.
 - Bens de consumo [4]: Incluindo empresas do setor de bebidas.

- Educação [1, 5 e 12]: Envolvendo instituições e empresas que oferecem serviços educacionais.
 - Financeiro [3 e 7]: Serviços bancários e outras soluções financeiras.
 - Varejo e atacado [11 e 13]: Empresas que atuam em comércio direto com o consumidor e no mercado atacadista.
 - Business-to-Government (B2G) [10]: Empresas que oferecem soluções direcionadas a governos e órgãos públicos.
- Especificidades das Empresas: Além dos setores, algumas empresas apresentaram características particulares.
 - Empresas em plena expansão [10], que buscam consolidar sua presença no mercado.
 - Organizações que atendem a múltiplas áreas [11], como varejo, financeiro e bancário, evidenciando um portfólio diversificado de atuação.
 - Multinacionais com presença global [4], destacando a importância das práticas de qualidade em ambientes altamente competitivos.

5.5.6 Tipos de Projetos/Produtos que estejam trabalhando:

Os projetos e produtos desenvolvidos pelos profissionais entrevistados refletem a diversidade de desafios e demandas enfrentadas pela área de tecnologia. Os resultados apontam para a atuação em diferentes frentes, abrangendo soluções financeiras, plataformas corporativas, sistemas de integração e digitalização de processos. As categorias principais identificadas são descritas a seguir:

- Projetos Financeiros [3, 5 e 11]: Muitos dos entrevistados trabalham em soluções voltadas para o setor financeiro, destacando a relevância desse segmento na adoção de métricas de qualidade e automação.
 - Soluções para crédito e cobrança: Desenvolvimento de sites para solicitação e gerenciamento de crédito (educacional e financeiro).
 - Produtos financeiros complexos: Incluindo sistemas que funcionam como bancos internos para clientes, plataformas de marketplace de benefícios e sistemas de gestão financeira.
 - ERP financeiros: Voltados ao planejamento e controle de recursos financeiros em diversos setores.

- Sistemas Corporativos e de Integração [2, 9 e 10]: Há um forte foco em soluções que promovem a eficiência operacional e a integração de processos empresariais.
 - Sistemas ERP especializados: Incluindo plataformas voltadas para o agronegócio e integração de produtos e processos.
 - Integração de processos (iPaaS): Conectores e tecnologias que integram sistemas diversos, como no projeto para distribuição de gás com suporte a múltiplos sistemas e integrações.
 - Sistemas modulares: Projetos de longo prazo para substituir processos manuais (ex.: desenvolvimento de um sistema para uma agência reguladora).
- Digitalização e Automação [4, 6 e 8]: Os entrevistados destacaram o impacto da digitalização em seus projetos.
 - Automação de processos de vendas: Facilitando a autonomia de representantes e aumentando a eficiência das operações comerciais.
 - Plataformas multiplataforma: Soluções que operam tanto em ambiente web quanto em aplicativos móveis, com foco em experiência do usuário e feedback contínuo.
 - Sistemas para grandes varejistas: Otimização do planejamento de estoque para equilibrar oferta e demanda.
- Desenvolvimento de Produtos e Suporte Evolutivo [8, 10 e 13]: Muitos profissionais também trabalham na evolução e manutenção de produtos existentes.
 - Sustentação de sistemas robustos e consolidados: Como no caso de um ERP com 19 anos de mercado que exige melhorias constantes para acompanhar a evolução tecnológica.
 - Projetos longos e dinâmicos: Como o sistema funerário para uma grande capital e a sustentação de sistemas administrativos.
 - Correções e melhorias: Garantia de qualidade e resolução de impedimentos para atender demandas dos clientes.
- Soluções Específicas por Setor [6, 10 e 13]: Projetos direcionados a setores específicos também foram destacados.
 - Ramo educacional: Desenvolvimento de plataformas de treinamento para colaboradores, com gestão de treinamentos obrigatórios e opcionais.
 - Saúde e bem-estar: Criação de marketplaces de benefícios.
 - Distribuição de gás: Sustentação e evolução de sistemas integrados.
- Soluções Focadas em Mobile e Web [4, 7 e 12]: Um dos temas recorrentes foi o foco no desenvolvimento de plataformas mobile (Android e iOS) e sites.

- Atender clientes finais com produtos intuitivos e acessíveis.
- Oferecer ferramentas práticas para colaboradores e representantes comerciais.

5.5.7 Experiência com Garantia da Qualidade de Produto:

Os profissionais entrevistados possuem uma ampla gama de experiências na área de Garantia da Qualidade, variando tanto no tempo de atuação quanto nos papéis e atividades desempenhados. Os principais aspectos observados podem ser categorizados da seguinte forma:

- Tempo de Experiência: Os entrevistados apresentam diferentes níveis de experiência, que variam de recém-ingressantes na área a profissionais altamente experientes.
 - Até 3 anos de experiência [1, 3, 4, 7, 9 e 12]: Inclui profissionais com 1 ano e meio, 2 anos e meio e 3 anos de atuação.
 - Entre 4 e 7 anos de experiência [2, 5, 8 e 10]: Contempla profissionais com 4, 5 anos e 3 meses, e 6 anos de atuação.
 - Mais de 10 anos de experiência [6, 11 e 13]: Abrange profissionais com 10, 13 e até 15 anos de atuação na área de QA, indicando uma trajetória consolidada.
- Papel e Responsabilidades: Os papéis desempenhados pelos entrevistados refletem uma evolução nas atividades realizadas e nas responsabilidades assumidas.
 - Testes manuais [1, 3, 6, 9, 11, 12 e 13]: Muitos profissionais começaram a carreira trabalhando com execução e escrita de casos de teste manuais.
 - Automação de testes [2, 3, 4, 5, 6 e 9]: Alguns entrevistados mencionaram experiência inicial com ferramentas como TestComplete e Selenium, priorizando testes regressivos e automação em processos como migração de dados entre bancos.
 - Liderança informal [6, 10 e 13]: Apesar de não ocuparem cargos formais de liderança, alguns assumem responsabilidades como organização de processos de teste, implantação de ferramentas e orientação de colegas.
 - Scrum Master [6]: Um dos entrevistados destacou a atuação como o primeiro Scrum Master do time de testes, liderando a implantação de metodologias ágeis.
 - Pseudo QA Lead [11]: Outro profissional relatou atuar como uma espécie de líder de QA, desempenhando funções estratégicas no time.
- Atribuições Relevantes: As atividades relatadas destacam a abrangência do trabalho na área de Qualidade.

- Execução de testes [2, 11 e 13]: Inclui a escrita e execução de casos de teste, desenvolvimento de automações iniciais e priorização de testes regressivos.
 - Gestão de qualidade no ciclo de desenvolvimento [8]: Um tema recorrente foi a importância de incorporar práticas de qualidade em todas as etapas do ciclo de desenvolvimento de produtos.
 - Apoio à equipe e orientação de colegas [6, 9, 10 e 12]: Experiências em atividades colaborativas, como suporte técnico e orientação sobre automação.
- Trajetórias Profissionais e Avanços: Para muitos profissionais, a experiência na Qualidade de Produto foi um trampolim para alcançar posições de maior responsabilidade.
 - Um entrevistado [10] destacou que a base em qualidade de produto foi fundamental para se tornar líder de projeto.
 - Outros [6, 11 e 13] mencionaram como a experiência em automação e testes manuais proporcionou oportunidades para atuar em contextos variados, desde migração de dados até gestão de qualidade em projetos complexos.
 - Áreas Emergentes e Ingressos Recentes: Um participante [2] destacou que a área de Qualidade de Produto é uma área relativamente nova em suas organizações, com desafios associados à implementação de processos e ferramentas adequados. Profissionais [1 e 12] com menos de 3 anos de experiência relataram atividades como aprendizado e implementação gradual de práticas de qualidade.

5.5.8 Experiência com Automação de Teste:

Os dados obtidos nas entrevistas demonstram uma ampla variedade de experiências e níveis de envolvimento dos profissionais entrevistados com automação de testes. A seguir, são apresentadas as principais categorias identificadas a partir das respostas:

- Diversidade de Experiência em Automação: Os níveis de experiência em automação de testes variam significativamente entre os entrevistados.
 - Iniciantes [1, 6, 8, 9, 11 e 12]: Alguns participantes relataram pouca experiência com automação, destacando fatores como receio técnico e predomínio de testes manuais em seus ambientes de trabalho. Um entrevistado, por exemplo, classificou seu nível de automação como 4/10, indicando uso limitado em projetos reais.
 - Experiência moderada [3, 4 e 13]: Profissionais com até dois anos de experiência relataram uso inicial de ferramentas como Selenium, Postman e Robot

Framework, com foco em aprendizado prático e integração de automação a processos manuais.

- Especialistas [2, 5 e 10]: Entrevistados com experiência consolidada mencionaram trabalho com múltiplas linguagens de programação e frameworks variados, incluindo Cypress, TestComplete, Appium e JMeter. Alguns têm mais de 10 anos de experiência com automação.
- Ferramentas e Tecnologias Mais Utilizadas: As ferramentas citadas refletem uma ampla gama de escolhas tecnológicas, utilizadas conforme as necessidades dos projetos.
 - Frameworks e linguagens [2, 3, 4, 5 e 12]: Cypress, Selenium, Robot Framework, Appium, TestComplete e JUnit foram amplamente mencionados.
 - Ferramentas de API e back-end [1, 3 e 5]: Postman, Hash Assured e PubTears aparecem em testes de APIs e automação de processos de back-end.
 - Testes avançados [10]: JMeter foi usado para testes de carga e estresse, garantindo a resiliência dos sistemas.
 - Novas tecnologias [13]: A automação em contextos mais recentes envolveu Flutter e Appian, mostrando a adaptação às demandas tecnológicas.
- Atividades Realizadas e Contribuições Relevantes: As funções e contribuições destacadas incluem os seguintes exemplos.
 - Automação de tarefas repetitivas [5 e 13]: Implementação de automação para otimizar testes manuais.
 - Criação de arquiteturas e POCs (Provas de Conceito): Um entrevistado [6] relatou o desenvolvimento de uma arquitetura de testes utilizada por toda a equipe.
 - Certificações específicas [10 e 13]: Alguns profissionais participaram de certificações técnicas, como habilitação de dispositivos de pagamento multibandeira (Visa e MasterCard).
 - Liderança técnica [2, 6 e 11]: Alguns participantes assumiram papéis como DevOps ou lideranças informais, auxiliando a equipe na organização de processos e ferramentas de automação.
- Desafios Enfrentados [1, 2, 4, 6, 11 e 13]: Os desafios relatados reforçam as complexidades da automação. Foram mencionados exemplos relacionados a limitações técnicas. Além de que a instabilidade de ambientes e adaptação a novas tecnologias também foram citadas como barreiras frequentes. Em alguns casos, a automação enfrentou dificuldades devido à redução de pessoal. Como também exemplos relacionados a curva de aprendizado onde a escolha de ferramentas como Robot Framework, com abstração mais acessível, foi mencionada como estratégia para superar limitações técnicas iniciais.

- **Evolução e Adaptação:** Os entrevistados apontaram trajetórias de crescimento e adaptação à automação.
 - De testes manuais para automação [2, 4, 6, 8, 9, 11 e 12]: Muitos profissionais começaram com QA manual, migrando gradualmente para a automação.
 - Experiências diversificadas [2, 5 e 13]: Alguns relataram trabalho com automação em múltiplos contextos, adaptando-se a diferentes tecnologias e tipos de produto.
 - Integração de habilidades [6, 12]: O conhecimento em QA manual foi usado para criar automações consistentes, garantindo a confiabilidade dos processos.

5.5.9 **Experiência marcante com a automação de teste:**

Os entrevistados relataram diversas experiências marcantes com automação no desenvolvimento de software, destacando desafios técnicos, aprendizados e impactos positivos tanto para as organizações quanto para o desenvolvimento profissional.

- **Transição para Automação e Aprendizados Iniciais:** Para muitos profissionais, a introdução à automação foi marcada por um período de aprendizado intenso, com apoio de mentorias e colaboração em equipe.
 - Aprendizado com mentores [1, 2, 8 e 11]: Alguns relataram a importância de mentorias internas e orientação de colegas mais experientes, especialmente no início da carreira.
 - Superação de resistências iniciais [2, 5, 6, 11 e 12]: A automação foi descrita como um desafio pessoal para alguns, que enfrentaram resistência inicial, mas superaram obstáculos ao aprenderem na prática e observarem resultados concretos.
 - Marco inicial inspirador [12]: Foi citado sobre o momento em que viu o primeiro teste automatizado ser executado com sucesso foi especialmente gratificante, simbolizando a superação de barreiras pessoais e técnicas.
- **Desafios Técnicos e de Negócio:** A automação frequentemente exigiu a superação de barreiras técnicas e organizacionais.
 - Configuração de pipelines e ferramentas [3, 4 e 10]: Desafios com ferramentas como Postman, integração manual de sistemas e configuração de pipelines foram comuns.
 - Adaptação às restrições do cliente [2, 3 e 6]: Muitos profissionais destacaram a necessidade de adaptar soluções às especificidades e limitações dos clientes, incluindo reestruturação manual e desenvolvimento de novos frameworks.

- Migração tecnológica [5, 6 e 8]: Projetos envolvendo a reescrita de sistemas em novas linguagens e tecnologias exigiram mentorias internas e colaboração com especialistas.
- Impacto Organizacional e Colaboração em Equipe: As experiências destacaram o impacto positivo da automação nos processos organizacionais.
 - Soluções para problemas críticos [5, 6, 8 e 13]: A automação foi vista como essencial para resolver dificuldades de regressões manuais e migração de dados entre bancos de dados.
 - Colaboração interna e externa [3, 6, 8, 9, 10, 11 e 13]: Projetos marcantes envolveram integração entre equipes de consultoria e clientes, superando barreiras organizacionais e agregando valor aos processos.
 - Formação de times especializados [6, 8 e 10]: Algumas empresas investiram na criação de equipes dedicadas à automação e promoveram treinamentos, como oficinas de JUnit, para elevar o nível técnico da equipe.
- Estruturação e Organização: A organização do trabalho foi citada como um fator crítico de sucesso na automação.
 - Uso de padrões como PageObject [9]: Estruturas organizadas facilitaram a adaptação e o entendimento do código, especialmente para iniciantes.
 - Desenvolvimento de frameworks internos [5]: A criação de frameworks robustos foi destacada como um marco, permitindo que as soluções automatizadas fossem replicáveis e sustentáveis no longo prazo.
- Evolução Profissional e Estratégica: As experiências marcantes também impactaram positivamente o desenvolvimento pessoal e profissional dos entrevistados.
 - De operacional para estratégico [5]: A automação elevou o nível de atuação de um dos profissionais, permitindo uma transição de tarefas operacionais para papéis mais estratégicos.
 - Fôlego renovado [6 e 11]: Para profissionais experientes em QA manual, aprender e aplicar automação trouxe um novo entusiasmo e sensação de renovação na carreira.
 - Reconhecimento [12]: Para um dos entrevistados o reconhecimento por trabalhos apresentados à liderança, destacando o impacto positivo da automação nos projetos e na organização.

5.5.10 Familiaridade com o conceito de qualidade:

As entrevistas revelaram uma ampla variação nos níveis de familiaridade com o conceito de qualidade e nas formas como os profissionais o interpretam e aplicam em suas práticas. Uma síntese das Perspectivas sobre Qualidade pode ser representada pela Figura 5.5:

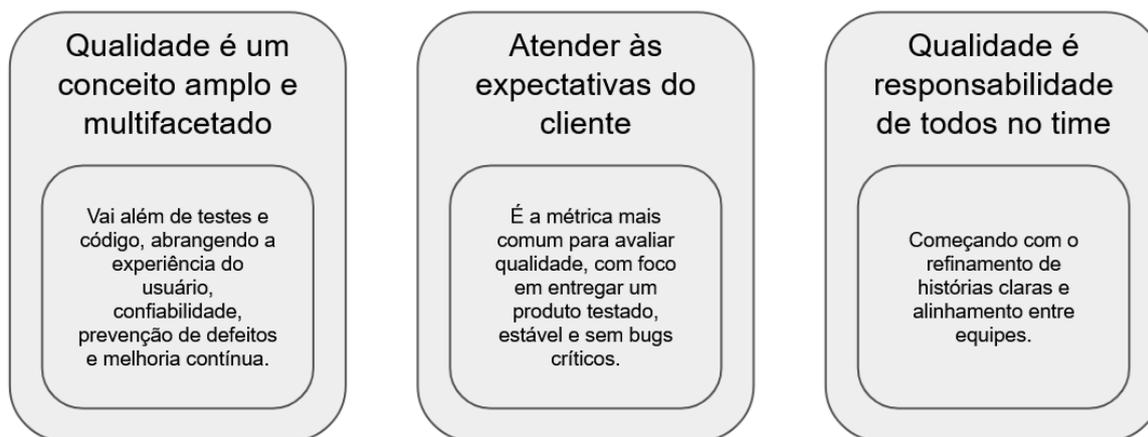


Figura 5.5 – Síntese das Perspectivas sobre Qualidade - segundo a indústria de TI

As perspectivas são relatadas com mais detalhes nas categorias abaixo indicadas.

- **Níveis de Familiaridade com o Conceito de Qualidade:** Os níveis de familiaridade relatados pelos entrevistados variaram entre baixo, mediano e alto.
 - Baixo nível de familiaridade [1 e 12]: Alguns profissionais demonstraram dificuldade em articular o conceito de qualidade, limitando-se a definições genéricas como a dependência de requisitos do cliente.
 - Nível mediano de familiaridade [6, 7, 9, 10, 11 e 13]: Entrevistados nesse grupo reconheceram aspectos amplos da qualidade, como a importância de atender expectativas do cliente e validar critérios técnicos, mas sem uma compreensão completamente abrangente.
 - Alto nível de familiaridade [2, 3, 4, 5 e 8]: A maioria dos profissionais afirmou ter alta familiaridade com qualidade, abordando conceitos avançados como prevenção de defeitos, melhoria contínua e a responsabilidade compartilhada entre as equipes.
- **Compreensões Fundamentais de Qualidade:** Os entrevistados trouxeram interpretações variadas sobre o que é qualidade.

- Atender às expectativas do cliente [1, 3, 4, 8, 9, 10, 11, 12 e 13]: Qualidade é amplamente vista como a entrega de produtos ou funcionalidades que atendam ou superem as necessidades do cliente.
- Eficiência e redução de retrabalho [2, 3 e 13]: Muitos destacaram a importância de evitar problemas e defeitos, garantindo processos mais eficazes.
- Amplo escopo do conceito [10]: Qualidade foi descrita como um conceito abrangente que vai além de testes, abrangendo aspectos técnicos, experiência do usuário e o ciclo de desenvolvimento como um todo.
- Satisfação do usuário final [3, 5, 9 e 10]: Qualidade também está associada à experiência do usuário e ao nível de confiança que o software proporciona.
- Aspectos Práticos e Aplicados de Qualidade: Os profissionais também destacaram aspectos práticos de qualidade no desenvolvimento de software.
 - Verificação e validação [6]: Dividir a qualidade em etapas que garantem conformidade com os requisitos e validação de que o produto atende às expectativas.
 - Metrificação e critérios claros [7]: A importância de medir qualidade por critérios e métricas definidas, como estabilidade, ausência de bugs críticos e adequação às necessidades do cliente.
 - Responsabilidade compartilhada [2, 6, 9, 12 e 13]: Qualidade não é vista apenas como responsabilidade dos QAs, mas de todo o time, começando pelo refinamento das histórias de usuário e pela colaboração interdisciplinar.
- Desafios e Percepções Subjetivas sobre Qualidade: Alguns entrevistados destacaram desafios e aspectos subjetivos relacionados à qualidade.
 - Conflito de prioridades [4 e 13]: A dificuldade de equilibrar as expectativas de times de produto e as necessidades do cliente foi mencionada como um obstáculo frequente.
 - Subjetividade do conceito [8]: Qualidade pode variar conforme o contexto do projeto e as expectativas dos stakeholders, tornando o conceito mais desafiador de padronizar.
 - Histórias mal refinadas [13]: Falhas no refinamento de histórias de usuário podem prejudicar o entendimento de qualidade, gerando retrabalho e inconsistências no desenvolvimento.

5.5.11 Familiaridade com o conceito de métricas:

As entrevistas mostraram uma ampla variação na familiaridade dos profissionais com métricas e suas aplicações no contexto de qualidade e desenvolvimento de software. Uma síntese das Perspectivas sobre Métricas, seus benefícios e desafios, pode ser representada pela Figura 5.6:



Figura 5.6 – Síntese das Perspectivas sobre Métricas - segundo a indústria de TI

As perspectivas são relatadas com mais detalhes nas categorias abaixo indicadas.

- Níveis de Familiaridade com Métricas: Os entrevistados demonstraram diferentes níveis de familiaridade, variando entre baixo, mediano e alto.
 - Baixo nível de familiaridade [6, 11 e 12]: Alguns profissionais relataram pouco envolvimento com métricas, com a responsabilidade frequentemente delegada a outros. Apesar disso, reconhecem métricas como instrumentos importantes para avaliar a qualidade do software.
 - Nível mediano de familiaridade [1, 3, 4, 5, 7, 9, 10 e 13]: Esses profissionais entendem métricas como indicadores úteis para mensurar qualidade e progresso, mas apontam dificuldades em aplicá-las no dia a dia, especialmente em contextos ágeis, onde o foco em entregas pode dificultar a coleta e análise de dados.

- Alto nível de familiaridade [2 e 8]: Muitos relataram alta familiaridade, usando métricas como ferramentas essenciais para priorizar atividades, medir progresso, identificar gargalos e embasar decisões estratégicas.
- Compreensão do Conceito de Métricas: Os entrevistados apresentaram diferentes interpretações sobre o que são métricas.
 - Indicadores de progresso e qualidade [4]: Métricas são vistas como formas de medir o estado atual de um projeto, avaliando conformidade com requisitos e identificando áreas de melhoria.
 - Ferramentas adaptáveis [7 e 13]: Métricas devem ser escolhidas com base nos objetivos do software e nas necessidades do cliente, variando de acordo com o contexto e estratégia de trabalho.
 - Instrumentos de comparação [8]: Algumas respostas destacaram a importância de métricas como padrões de comparação, permitindo análise de desempenho em relação a metas predefinidas.
- Aplicações Práticas e Benefícios das Métricas: As métricas foram destacadas como essenciais com o objetivo de alcançar os seguintes objetivos.
 - Priorização e melhoria contínua [10]: Métricas ajudam a identificar gargalos, evoluir processos e concentrar esforços em aspectos críticos do projeto.
 - Comunicação com stakeholders [12]: Indicadores visuais, como dashboards e gráficos (ex.: burn down, burn up), são usados para atualizar clientes e equipes sobre o progresso do projeto, especialmente em contratos de longo prazo.
 - Identificação de problemas [13]: Métricas como bugs por release e incidentes em produção foram mencionadas como importantes para identificar padrões e áreas problemáticas, permitindo ajustes no processo.
- Desafios na Implementação de Métricas: Foram indicados alguns exemplos de desafios na utilização de métricas.
 - Subutilização de métricas [4 e 6]: Alguns relataram que métricas de qualidade são pouco usadas em suas experiências, o que pode levar a interpretações erradas ou lacunas no controle de qualidade.
 - Dificuldades no contexto ágil [6]: O foco em entregas ao final de sprints frequentemente adia os testes e dificulta a coleta de métricas, prejudicando a análise de qualidade em tempo real.
 - Complexidade na adaptação [4]: A escolha de métricas varia conforme o projeto, o que exige flexibilidade e alinhamento com objetivos específicos do cliente e do negócio.

5.5.12 Familiaridade com o conceito de automação de testes:

As entrevistas destacaram diferentes níveis de familiaridade e percepções sobre automação de testes, com foco nos benefícios, desafios e aplicações práticas. Uma síntese das Perspectivas sobre Automação de Testes, seus benefícios e desafios, pode ser representada pela Figura 5.7:



Figura 5.7 – Síntese das Perspectivas sobre Automação de Testes - segundo a indústria de TI

As perspectivas são relatadas com mais detalhes nas categorias abaixo indicadas.

- Níveis de Familiaridade com Automação de Testes: Os níveis de familiaridade relatados variaram entre baixo, mediano e alto.
 - Baixo nível de familiaridade [1 e 8]: Alguns entrevistados indicaram conhecimento limitado, com foco predominante em testes manuais. A automação foi mencionada como algo implementado esporadicamente, quando há tempo disponível.
 - Nível mediano de familiaridade [4, 6, 7, 10, 11, 12 e 13]: Profissionais com experiência inicial em automação destacaram seu valor, mas relataram limitações, como dependência de ferramentas específicas e a falta de infraestrutura para dispositivos iOS.

- Alto nível de familiaridade [2, 5 e 9]: Muitos entrevistados demonstraram domínio de automação, reconhecendo sua importância para aumentar a eficiência, reduzir esforços manuais e garantir confiabilidade no processo de desenvolvimento.
- Benefícios da Automação de Testes: Os entrevistados destacaram os seguintes benefícios.
 - Otimização de tempo e esforço [1, 4, 5, 8, 10, 11 e 12]: Automação reduz tarefas manuais repetitivas, como testes de regressão, poupando tempo e permitindo foco em atividades mais estratégicas.
 - Agilidade e confiabilidade [11]: Proporciona respostas rápidas sobre impactos de alterações no sistema, garantindo maior segurança na entrega de software.
 - Cobertura de testes [9]: Permite ampliar a cobertura de testes, assegurando que fluxos críticos e recorrentes sejam validados consistentemente.
 - Prevenção de problemas em produção [4 e 12]: Automação ajuda a identificar e corrigir falhas antes que o software seja disponibilizado aos usuários finais.
- Aplicações e Ferramentas de Automação: Os profissionais citaram diversas ferramentas e práticas utilizadas na automação.
 - Ferramentas mais mencionadas [10 e 13]: Selenium, Cypress, Appium e JMeter foram destacados como opções para diferentes tipos de testes, como funcionalidade, performance e carga.
 - Frameworks acessíveis [6]: O Robot Framework foi elogiado por sua curva de aprendizado mais baixa, permitindo que novos integrantes do time adotem a automação sem necessidade de amplo conhecimento em programação.
 - Aplicações práticas [12]: Exemplos incluem automação para criação de usuários recorrentes, validação de fluxos críticos e execução de testes em produção.
- Desafios na Implementação de Automação de Testes: Apesar dos benefícios, os entrevistados mencionaram desafios comuns na automação.
 - Falta de foco e tempo [11 e 12]: Em alguns projetos, a automação é relegada a segundo plano devido a prazos apertados ou foco em entregas rápidas.
 - Infraestrutura específica [6]: Automação para dispositivos iOS requer infraestrutura especializada (como Macs e iPhones), o que pode limitar a implementação.
 - Avaliação de custo-benefício [10]: É necessário priorizar cenários que gerem maior valor ao projeto, evitando esforços em áreas com pouco impacto.
- Estratégias e Abordagens para Automação: Os profissionais destacaram a importância de estratégias claras para maximizar o valor da automação.

- Identificar "dores" do projeto [13]: Antes de automatizar, é essencial entender os problemas mais críticos que precisam ser resolvidos.
- Equilíbrio entre esforço e impacto [11]: Priorizar a automação de funcionalidades principais e fluxos críticos para balancear o custo com os benefícios obtidos.
- Integração com testes manuais [13]: O domínio de testes manuais e do negócio foi considerado fundamental antes de partir para automação.

5.5.13 Principais / Mais importantes Métricas:

As entrevistas revelaram uma variedade de métricas utilizadas pelos profissionais para monitorar e garantir a qualidade de produtos de software. Essas métricas abrangem desde aspectos técnicos, como bugs e desempenho, até questões de experiência do usuário e valor para o negócio. Sínteses das Principais ou Mais Importantes Métricas de Qualidade de Produto podem ser representadas pela Figura 5.8, desmembradas também nas Figuras 5.9, 5.10 e 5.11:

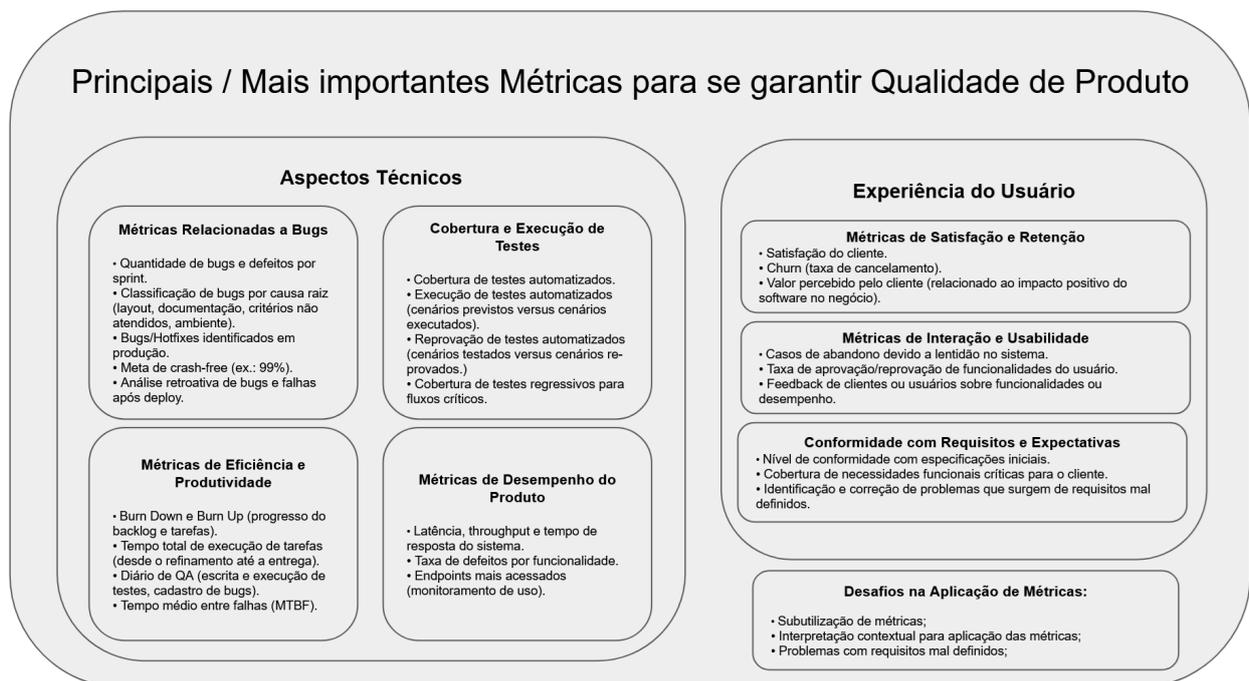


Figura 5.8 – Principais / Mais Importantes Métricas de Qualidade de Produto



Figura 5.9 – Aspectos Técnicos das Métricas Identificadas

As métricas são relatadas com mais detalhes abaixo:

- Aspectos Técnicos: Essas métricas estão diretamente relacionadas ao desempenho, estabilidade e eficiência técnica do software, bem como à qualidade do processo de desenvolvimento.
 - Métricas Relacionadas a Bugs [1, 2, 3, 4, 5, 6, 8, 10, 11 e 13]:
 - * Quantidade de bugs e defeitos por sprint.
 - * Classificação de bugs por causa raiz (layout, documentação, critérios não atendidos, ambiente).
 - * Bugs/Hotfixes identificados em produção.
 - * Meta de crash-free (ex.: 99%).
 - * Análise retroativa de bugs e falhas após deploy.
 - Cobertura e Execução de Testes [3, 5, 9 e 12]:
 - * Cobertura de testes automatizados.

- * Execução de testes automatizados (cenários previstos versus cenários executados).
 - * Reprovação de testes automatizados (cenários testados versus cenários reprovados.)
 - * Cobertura de testes regressivos para fluxos críticos.
- Métricas de Eficiência e Produtividade [8, 10, 11 e 13]:
- * Burn Down e Burn Up (progresso do backlog e tarefas).
 - * Tempo total de execução de tarefas (desde o refinamento até a entrega).
 - * Diário de QA (escrita e execução de testes, cadastro de bugs).
 - * Tempo médio entre falhas (MTBF).
- Métricas de Desempenho do Produto [7 e 9]:
- * Latência, throughput e tempo de resposta do sistema.
 - * Taxa de defeitos por funcionalidade.
 - * Endpoints mais acessados (monitoramento de uso).

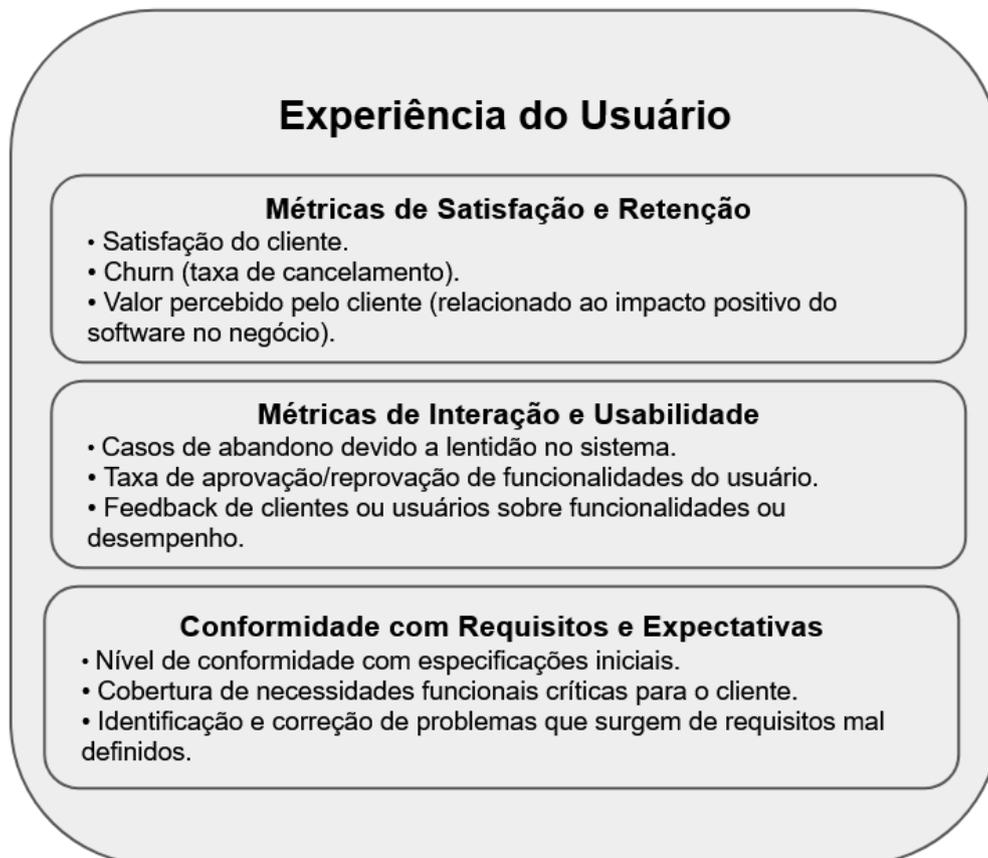


Figura 5.10 – Aspectos de Experiência do Usuário das Métricas Identificadas

- Experiência do Usuário: Estas métricas avaliam como o software atende às expectativas e necessidades dos usuários e clientes finais.

- Métricas de Satisfação e Retenção [8]:
 - * Satisfação do cliente.
 - * Churn (taxa de cancelamento).
 - * Valor percebido pelo cliente (relacionado ao impacto positivo do software no negócio).
- Métricas de Interação e Usabilidade [9 e 12]:
 - * Casos de abandono devido a lentidão no sistema.
 - * Taxa de aprovação/reprovação de funcionalidades no contexto do usuário.
 - * Feedback direto de clientes ou usuários sobre funcionalidades ou desempenho.
- Conformidade com Requisitos e Expectativas [6 e 8]:
 - * Nível de conformidade com especificações iniciais.
 - * Cobertura de necessidades funcionais críticas para o cliente.
 - * Identificação e correção de problemas que surgem de requisitos mal definidos.

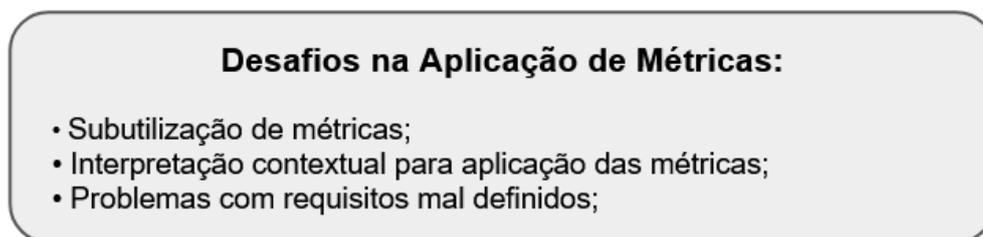


Figura 5.11 – Desafios na Aplicação de Métricas

- Desafios na Aplicação de Métricas [6 e 13]: Os profissionais também relataram dificuldades na implementação e uso eficaz de métricas.
 - Subutilização de métricas: Alguns times ainda não adotaram métricas de forma consistente.
 - Interpretação contextual: A quantidade de bugs deve ser analisada considerando o contexto e a experiência da equipe, evitando conclusões precipitadas.
 - Problemas com requisitos mal definidos: Podem gerar falhas desde o início do ciclo de desenvolvimento, impactando os resultados das métricas.

5.5.14 Como as Métricas são Utilizadas:

As entrevistas revelaram como as métricas são coletadas, utilizadas e analisadas nas organizações, destacando ferramentas, papéis e finalidades no processo de desenvolvimento de software.

- Coleta de Métricas: As métricas são coletadas por diversas ferramentas e processos, variando de abordagens automatizadas a manuais.
 - Ferramentas Automáticas [1, 2, 10 e 11]: Jira, Zephyr, Azure DevOps e Kanbanize são amplamente utilizadas para registrar informações, gerar relatórios e configurar dashboards automatizados.
 - Coleta Manual [10]: Em contextos com menos recursos tecnológicos, as métricas são registradas manualmente, muitas vezes em planilhas Excel com modelos pré-configurados.
 - Fontes de Dados [7, 8 e 9]: Métricas são extraídas de chamados e incidentes, feedbacks de usuários, registros de bugs e bloqueios de tarefas, além de dados técnicos (como tempo de carregamento de telas).
 - Tagueamento e diários de uso [8]: Detalhes sobre interações dos usuários com o produto ajudam a capturar dados ricos para análises futuras.
- Papéis e Responsabilidades: A coleta e análise de métricas envolvem diferentes membros do time.
 - Engenheiros de Software (DEVs) [7]: Focam em métricas técnicas, como desempenho, latência e tempo de resposta.
 - Analistas de Negócio (BAs) [7]: Avaliam métricas relacionadas a custos, preços e impacto no negócio.
 - Product Managers, Technical Managers e Tech Leads [7]: Centralizam as métricas e as utilizam para guiar decisões estratégicas, como definição de prioridades no ciclo de desenvolvimento.
 - Coordenadores e liderança técnica [1, 3, 4, e 12]: São responsáveis por consolidar as métricas e apresentar os resultados para o time.
 - Time como um todo [2 e 12]: Registra métricas e as discute em retrospectivas e revisões, promovendo colaboração e melhoria contínua.
- Finalidades e Análises: As métricas têm aplicações variadas no ambiente de trabalho.
 - Monitoramento e Feedback Contínuo [5 e 8]: Ferramentas como dashboards oferecem visibilidade sobre progresso, desempenho e impedimentos, permitindo ajustes rápidos.

- Identificação de Problemas e Causas Raiz [2 e 6]: Métricas ajudam a localizar falhas no ciclo de desenvolvimento, como problemas na escrita de histórias, desenvolvimento ou homologação.
 - Acompanhamento de Progresso [10]: Indicadores como Burn Down e Burn Up são usados para medir a evolução do backlog e prever prazos.
 - Análise em Retrospectivas [2 e 11]: Times revisam métricas para identificar padrões de bugs, desempenho do projeto e áreas que precisam de melhorias.
 - Avaliação de Impacto das Melhorias [9 e 11]: Após ajustes no produto, novas métricas são coletadas para avaliar a eficácia das mudanças implementadas.
- Processos e Práticas Relacionadas às Métricas: As métricas estão integradas a processos de gestão e desenvolvimento.
 - Shift Left Testing [5]: Escrita de testes antes do desenvolvimento para prevenir falhas futuras.
 - Obrigações e Transparência [12]: Registro de métricas é obrigatório para avançar tarefas, e sua apresentação promove alinhamento entre o time e os stakeholders.
 - Revisões Periódicas [9, 10 e 11]: Em alguns casos, métricas são revisadas trimestralmente em boards específicos.
 - Apoio à Decisão [2, 3 e 10]: Dados são usados para guiar decisões baseadas em evidências, alinhando expectativas de clientes e equipes.
 - Desafios e Observações: Apesar dos benefícios, algumas dificuldades foram mencionadas.
 - Dependência de Ferramentas [10]: A falta de ferramentas adequadas pode limitar a coleta automatizada, forçando registros manuais que consomem mais tempo.
 - Interpretação de Dados [7, 8, 9, 10, 11 e 13]: Métricas precisam ser analisadas em contexto para evitar conclusões equivocadas.
 - Resistência do Times [10 e 13]: Em alguns ambientes, o uso de métricas pode ser percebido como uma ferramenta de avaliação hierárquica, gerando receios.

5.5.15 Como as Métricas são Definidas:

As entrevistas mostraram que o processo de definição e revisão de métricas varia conforme o contexto organizacional e o papel desempenhado por diferentes membros da equipe.

- Processo de Definição das Métricas: As métricas são geralmente definidas de forma colaborativa, com a participação de diversos papéis na equipe, embora em alguns casos sejam impostas por lideranças ou clientes.
 - Colaboração Multidisciplinar [7]:
 - * As métricas são definidas no discovery do produto, com a participação de Product Managers, Technical Managers, engenheiros e líderes de QA.
 - * O Product Manager frequentemente define a North Star Metric (métrica central) do produto, a partir da qual outras métricas são derivadas.
 - Liderança e Papéis Específicos [4, 5, 10 e 13]:
 - * Em muitos casos, os líderes de projeto, Scrum Masters ou gerentes de testes definem as métricas iniciais e as apresentam ao time.
 - * O líder incentiva o engajamento do time, reforçando a importância das métricas para visibilidade e tomada de decisão.
 - Definições Baseadas em Experiência ou Cliente [2, 3, 9 e 11]:
 - * Algumas métricas, especialmente de performance, são definidas diretamente pelos QA, com base em sua experiência prática.
 - * Em ambientes onde o cliente dita os critérios de qualidade, as métricas são definidas externamente, e a equipe segue rigorosamente os padrões estabelecidos.
- Revisão e Evolução das Métricas: A frequência e abordagem para revisar as métricas variam, com algumas organizações adotando revisões estruturadas e outras revisando métricas apenas quando necessário.
 - Revisões Periódicas [2, 3, 5, 7, 8 e 9]:
 - * Em muitas organizações, as métricas são revisadas trimestralmente ou mensalmente, como parte de um ciclo contínuo de melhoria.
 - * A revisão considera a relevância e eficácia das métricas, buscando alinhar-se às prioridades do projeto.
 - Revisões Baseadas em Problemas [4]:
 - * Algumas equipes revisam métricas apenas quando surgem problemas significativos ou quando identificam gaps em retrospectivas.
 - * Essas revisões são usadas para ajustar métricas ou criar novas, com base em aprendizados acumulados.
 - Falta de Revisão Formal [7, 10, 11 e 13]:
 - * Em alguns casos, as métricas seguem um modelo padronizado que raramente é revisado. Ainda assim, existe flexibilidade para ajustes pontuais quando necessário.

- Ferramentas e Critérios de Definição: As ferramentas e critérios usados no processo de definição e coleta das métricas foram amplamente mencionados.
 - Ferramentas [3 e 8]:
 - * Jira, Zephyr, Azure DevOps e BI (Business Intelligence) são usados como plataformas para definir, coletar e monitorar métricas.
 - Critérios [7 e 8]:
 - * Antes da coleta, as métricas são definidas com clareza, incluindo objetivos, metas e checkpoints (ex.: critérios para mover cards entre estágios de desenvolvimento).
 - * Métricas técnicas, como latência e tempo de resposta, são consideradas fundamentais e geralmente não negociáveis.
- Envolvimento do Time: O nível de participação do time na definição de métricas varia conforme o contexto.
 - Participação Ativa [2, 3, 5, 7 e 9]: Em equipes que promovem abordagens colaborativas, todo o time contribui para criar ou ajustar métricas, promovendo maior compreensão e aceitação.
 - Participação Limitada [10 e 12]: Em alguns casos, o time operacional não participa diretamente na definição ou revisão das métricas, mas utiliza as informações geradas para guiar suas atividades.
 - Definição Exclusiva pela Liderança ou Cliente [4, 11 e 13]: Quando as métricas são estabelecidas por clientes ou líderes, a equipe segue padrões pré-definidos, sem autonomia para modificá-los.

5.5.16 **Papel da Automação de Testes na Coleta das Métricas:**

As entrevistas revelaram percepções diversificadas sobre o impacto da automação de testes na coleta de métricas, variando desde benefícios significativos até desafios associados ao contexto e à implementação.

- Benefícios da Automação na Coleta de Métricas: Os entrevistados apontaram diversos benefícios proporcionados pela automação na coleta e análise de métricas.
 - Objetividade e Confiabilidade [3, 6, 7 e 8]:
 - * A automação reduz o esforço manual, minimiza erros e oferece dados consistentes para análise.

- * Proporciona maior precisão, especialmente em testes de regressão, ao identificar falhas e padrões de erros de maneira objetiva.
- Monitoramento em Tempo Real [8]:
 - * Permite a geração contínua de relatórios e métricas em tempo real, facilitando a identificação rápida de problemas e o acompanhamento de indicadores-chave.
- Eficiência e Redução de Retrabalho [5, 8, 10, 12 e 13]:
 - * Diminui o retrabalho durante as sprints ao automatizar tarefas repetitivas, como retestes de funcionalidades previamente estáveis.
 - * Ajuda a garantir que funcionalidades críticas continuem funcionando após alterações no sistema.
- Capacidades Avançadas [7 e 10]:
 - * A automação é fundamental para realizar testes de estresse, carga e resiliência, capturando métricas específicas de desempenho e segurança.
- Contextos e Limitações: Embora os benefícios sejam amplos, os entrevistados também destacaram fatores que podem limitar a eficácia da automação na coleta de métricas.
 - Dependência do Contexto e Implementação [2]:
 - * Ambientes desatualizados entre testes e produção, ou práticas inadequadas como hard code waits, podem distorcer os resultados coletados pela automação.
 - * O impacto da automação depende do tipo de métrica utilizada, sendo mais eficaz em métricas técnicas e repetitivas.
 - Participação Limitada da Automação [4]:
 - * Em algumas equipes, a automação ainda não está plenamente integrada à coleta de métricas, com práticas predominantemente manuais.
 - * A automação é frequentemente aplicada após a entrega de uma sprint, limitando seu uso como ferramenta de suporte em tempo real.
- Casos de Uso Comuns da Automação: A automação é usada de forma ampla para apoiar a coleta de métricas em diversas áreas.
 - Testes Regressivos [5, 9 e 10]:
 - * Identificação de bugs funcionais e coleta de métricas de falhas durante o processo de regressão, com discussões posteriores em retrospectivas para ajustes.

- Métricas de Desempenho [7]:
 - * Coleta de dados como latência, throughput e tempo de resposta para identificar gargalos no sistema e avaliar a estabilidade do ambiente antes de novos lançamentos.
- Padrões de Qualidade [5 e 6]:
 - * A frequência de erros em testes automatizados fornece insights sobre oscilações na qualidade e sinaliza áreas que precisam de atenção adicional.
- Desafios e Considerações para Melhorar o Impacto: Os entrevistados também destacaram desafios e práticas para melhorar o impacto da automação na coleta de métricas.
 - Integração Gradual [11]:
 - * Em algumas organizações, há um movimento para ampliar o uso da automação, com a intenção de integrá-la completamente à coleta de métricas.
 - Foco em Ambientes Estáveis [13]:
 - * A automação durante a subida do código ajuda a garantir que o ambiente esteja estável, reduzindo perdas de tempo com falhas no ambiente de testes.
 - Prioridade para Testes Automatizados [6]:
 - * Testes automatizados devem ser priorizados em relação aos testes manuais para aumentar a confiabilidade e abrangência das métricas.

5.5.17 Lições Aprendidas / Benefícios que a automação traz para a Qualidade do Produto:

As entrevistas destacaram várias lições importantes que a automação de testes trouxe para a experiência dos profissionais na garantia da qualidade de produtos de software.

- Benefícios Operacionais e Estratégicos: A automação de testes trouxe melhorias significativas nos processos de desenvolvimento e qualidade.
 - Otimização de Tempo [1, 6, 8 e 10]:
 - * A automação reduziu o tempo necessário para executar cenários de teste repetitivos, como testes de regressão.
 - * Exemplos mencionaram a execução de 60 cenários em apenas 10 minutos, comparado ao tempo considerável dos testes manuais.

- Agilidade e Eficiência [3, 6 e 9]:
 - * A automação permitiu respostas rápidas durante o avanço do desenvolvimento e reduziu significativamente o retrabalho.
 - * Testes regressivos, anteriormente demorados, agora são realizados de forma ágil, permitindo foco em outras áreas.
- Maior Produtividade [6, 8 e 10]:
 - * Eliminação de tarefas manuais repetitivas e maior aproveitamento de recursos humanos em atividades estratégicas.
- Compreensão Ampliada e Melhor Colaboração: A automação proporcionou uma visão mais ampla dos produtos e maior colaboração entre equipes.
 - Perspectiva Ampliada [4]:
 - * Permitiu aos profissionais entender melhor os serviços e produtos, trazendo insights valiosos sobre otimizações possíveis.
 - * Testes manuais não oferecem a mesma profundidade de entendimento em certos casos, enquanto a automação evidenciou processos passíveis de otimização.
 - Colaboração entre Times [2]:
 - * A automação requer suporte do time de desenvolvimento, reforçando a importância de trabalho conjunto entre Devs e QA.
 - * Integrações com ferramentas como DataDog e Google Chat foram destacadas como formas de aumentar a colaboração e a eficácia.
- Aprendizados Técnicos e Estratégicos: A implementação de automação trouxe aprendizados sobre boas práticas e desafios técnicos.
 - Estruturação de Código e Planejamento [9 e 13]:
 - * A automação exige scripts bem estruturados e robustos, com investimento inicial em sua criação e manutenção.
 - * A avaliação do esforço vs. ganho foi considerada essencial para decidir quais cenários automatizar.
 - Gradualidade do Processo [5 e 13]:
 - * A automação foi descrita como uma estratégia de médio a longo prazo, onde resultados confiáveis levam tempo para serem alcançados.
 - * O processo geralmente começa com testes manuais, evoluindo gradualmente para a automação.
- Limitações e Equilíbrio com Testes Manuais: Apesar dos benefícios, os profissionais destacaram a importância de equilibrar automação com testes manuais.

- Testes Manuais Insuperáveis em Certos Contextos [4 e 9]:
 - * O olhar humano continua essencial para avaliar criticidade, percepção de problemas complexos e subjetivos.
- Seleção Criteriosa de Cenários [13]:
 - * Nem todos os sistemas ou cenários são adequados para automação, sendo necessário considerar a estabilidade e a duração dos testes.
- Impactos na Qualidade e nas Métricas: A automação impactou diretamente na qualidade e no monitoramento de métricas.
 - Geração de Dados Automatizada [3, 10 e 11]:
 - * A automação facilitou a criação de dados de teste (mocks e massas), permitindo validações em grande escala.
 - Acompanhamento de Qualidade [7 e 8]:
 - * A automação simplificou a coleta de métricas relacionadas a bugs, desempenho e resiliência do sistema.
 - * Relatórios gerados automaticamente serviram como base para discussões e decisões estratégicas.

5.5.18 Ferramentas de Execução ou Automação de Testes Utilizadas:

As entrevistas revelaram uma ampla variedade de ferramentas e tecnologias utilizadas pelos profissionais para execução e automação de testes, abrangendo diferentes propósitos e contextos no desenvolvimento de software. Uma síntese das ferramentas de execução ou automação de testes pode ser representada pela Figura 5.12:



Figura 5.12 – Ferramentas de Execução ou de Automação de Testes

As ferramentas são relatadas com mais detalhes abaixo:

- Ferramentas de Automação para Testes Funcionais: Estas ferramentas são amplamente utilizadas para automação de testes funcionais em diferentes camadas do sistema.
 - Front-end e Back-end [1, 2, 3, 5, 6, 8, 9, 10, 12 e 13]:
 - * Cypress: Destacada como uma das ferramentas mais populares para automação de testes tanto de front-end quanto de back-end, com uso frequente em validações de interface e APIs.
 - * Selenium: Amplamente utilizado com diferentes linguagens, como CSharp e JavaScript, para automação de testes de interface.
 - * Playwright: Reconhecida por sua flexibilidade e facilidade de integração com diversos projetos.
 - * Robot Framework: Elogiado por sua baixa curva de aprendizado e aplicação prática em múltiplos contextos.
 - BDD (Behavior-Driven Development) [13]:
 - * Cucumber com Selenium: Aplicado principalmente para testes baseados em cenários, utilizando linguagem Gherkin para facilitar a comunicação entre equipes.
- Ferramentas para Testes de API [2, 8, 11, 12 e 13]: As ferramentas para validação de APIs foram amplamente mencionadas, devido à sua importância para garantir a comunicação entre sistemas.

- Postman: A mais citada para testes de API, com uso combinado ao Newman para execução em linha de comando.
 - Rest Assured: Popular para validações automatizadas de APIs RESTful.
 - Swagger: Utilizado para inspecionar e testar APIs diretamente pela interface.
 - SoapUI: Ferramenta tradicional para testes de APIs SOAP e REST.
- Ferramentas para Testes de Performance e Carga [1, 10 e 13]: As ferramentas de performance foram destacadas como essenciais para avaliar a resiliência e capacidade dos sistemas.
 - JMeter: Amplamente utilizado para testes de carga e stress, avaliando limites do sistema.
 - TestComplete: Aplicado em cenários de performance mais específicos e detalhados.
- Ferramentas de Gestão e Organização de Testes: Estas ferramentas suportam o planejamento, rastreamento e organização de testes.
 - Jira com Zephyr [4]: Integradas para gerenciar casos de teste, defeitos e métricas. O Zephyr, em particular, foi elogiado por sua integração com IA, embora tenha sido mencionado como uma solução cara.
 - Kanbanize [11]: Utilizado para organização de projetos e fluxos de trabalho no desenvolvimento de software.
- Ferramentas de Integração e Infraestrutura [7 e 11]: Ferramentas que suportam a execução de testes em ambientes complexos e integrados.
 - Jenkins: Utilizado para integração contínua, suportando a execução automatizada de testes após cada build.
 - BrowserStack: Oferece suporte para testes em múltiplos navegadores e dispositivos, simulando diferentes ambientes.
 - AWS Explorers e Pods/Kubernetes: Suportam a execução de testes em ambientes distribuídos e escaláveis.
- Ferramentas e Linguagens Próprias [7]: Além das ferramentas amplamente conhecidas, alguns entrevistados mencionaram o uso de linguagens ou ferramentas desenvolvidas internamente, adaptadas às necessidades específicas de seus projetos.

5.5.19 Metodologias de Desenvolvimento Utilizadas:

As entrevistas revelaram que as organizações adotam uma variedade de metodologias de desenvolvimento de software, adaptando-as conforme a natureza dos projetos e as necessidades dos times. As principais abordagens identificadas incluem Scrum, Kanban e combinações híbridas, como Scrumban. Uma síntese das metodologias pode ser representada pela Figura 5.13:

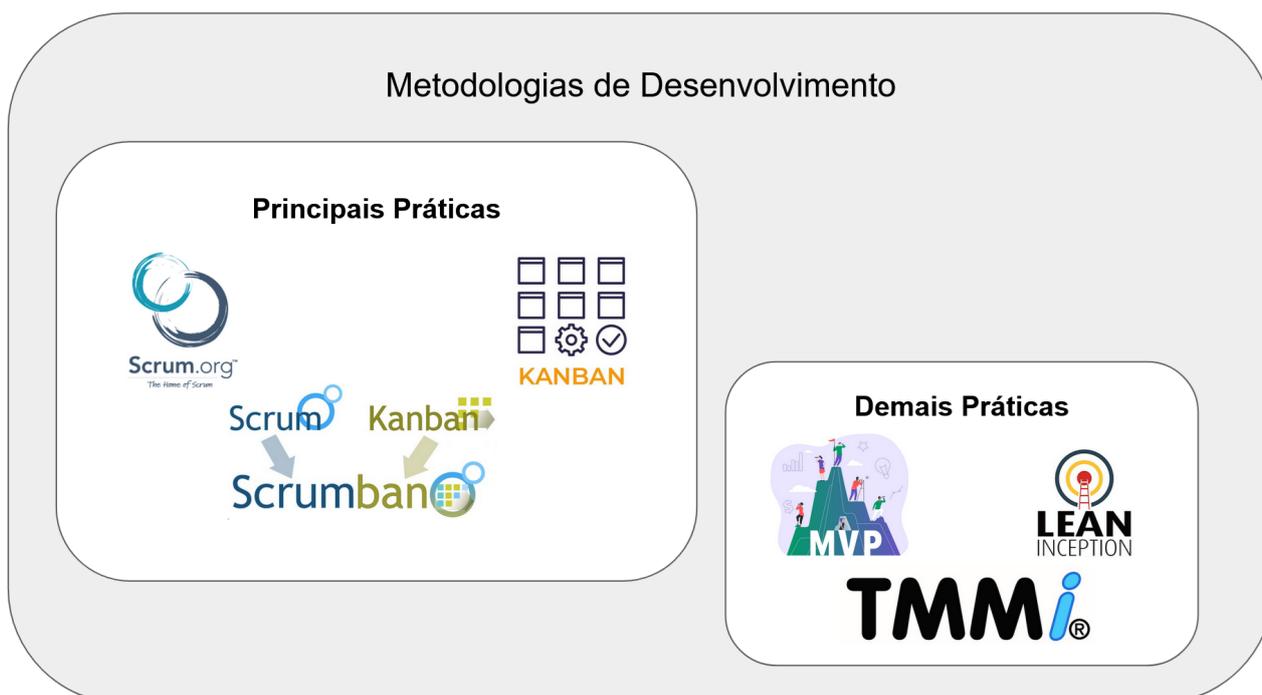


Figura 5.13 – Metodologias de Desenvolvimento

As metodologias são relatadas com mais detalhes abaixo:

- Scrum como Metodologia Ágil mais Utilizada: O Scrum foi amplamente citado como uma metodologia comum, embora com adaptações.
 - Implementações Típicas [1, 3, 4, 5, 6, 8, 9, 10, 12 e 13]:
 - * Uso de sprints com duração de duas semanas.
 - * Cerimônias tradicionais, como planejamento, revisão e retrospectivas.
 - Adaptações e Simplificações [2 e 7]:
 - * Muitas equipes utilizam o Scrum de forma simplificada, sem seguir o modelo "by the book".
 - * Exemplos incluem a ausência de Scrum Masters ou flexibilização de cerimônias.
 - Desafios Relatados [6 e 7]:

- * Problemas com timeboxes, previsibilidade e prazos curtos, que dificultam a execução do modelo clássico.
 - * Limitações na aplicação em projetos de manutenção evolutiva, onde as demandas não seguem ciclos bem definidos.
- Kanban com Foco em Fluxo Contínuo: O Kanban é frequentemente utilizado, especialmente em projetos de sustentação e automação.
 - Vantagens do Kanban [3, 5, 6 e 7]:
 - * Adequação ao fluxo contínuo de entregas e tarefas individuais, como automação de testes.
 - * Melhor alinhamento com dinâmicas de startups e projetos de sustentação.
 - Exemplos de Uso [8 e 10]:
 - * Gerenciamento de tarefas de automação em um modelo contínuo e visual.
 - * Sustentação de sistemas, onde as demandas chegam de forma imprevisível.
 - Combinações Híbridas como Scrumban e Flexibilidade: A combinação de Scrum e Kanban (Scrumban) foi destacada como uma prática que une o melhor de ambas as abordagens.
 - Características do Scrumban [4 e 7]:
 - * Organização do fluxo de trabalho com a flexibilidade do Kanban e o planejamento estruturado do Scrum.
 - Aplicação Dinâmica [8 e 10]:
 - * Escolha do framework mais adequado conforme o momento do projeto.
 - * Uso mais intenso de uma abordagem (Scrum ou Kanban) dependendo das necessidades do time.
 - Outras Abordagens e Ferramentas: Além de Scrum e Kanban, outras metodologias e práticas foram mencionadas.
 - MVP (Minimum Viable Product) [2]: Utilizado para validação de ideias em estágios iniciais de produtos.
 - Lean Inception [2]: Aplicado para alinhar expectativas e definir o escopo inicial de projetos.
 - TMMI (Test Maturity Model Integration) [2]: Utilizado em organizações que buscam maturidade em processos de teste.
 - Ferramentas de Gestão [9]: Azure DevOps foi mencionado como um suporte para gestão de tarefas em Scrum e Kanban.

- Flexibilidade e Adaptação [5, 6, 7, 8 e 10]: Uma abordagem prática e flexível foi destacada como essencial.
 - As empresas escolhem e adaptam metodologias conforme suas demandas e contextos específicos.
 - Combinações e ajustes são realizados para atender às dinâmicas de diferentes projetos, desde sustentação até desenvolvimento de novas funcionalidades.

5.5.20 Experiências de Insucesso nos Testes:

As entrevistas revelaram diversas situações de insucesso em testes de software, demonstrando que falhas podem ocorrer por motivos que vão além da responsabilidade do QA. Os temas abaixo foram agrupados para destacar as causas e lições aprendidas com essas experiências.

- Falhas por Planejamento Inadequado e Urgências: Uma causa comum de insucesso relatada foi a falta de planejamento adequado, especialmente em cenários de alta pressão.
 - Demandas de Última Hora [10]:
 - * Solicitações urgentes do cliente frequentemente levaram a testes apressados, sem tempo para cobrir todos os cenários necessários.
 - * Insuficiência de tempo para planejamento gerou impactos em qualidade, demandando correções posteriores e retrabalho.
 - Planejamento Deficiente [8 e 9]:
 - * A falta de uma visão completa dos impactos de novas funcionalidades, especialmente em sistemas com muitas dependências, resultou em falhas significativas.
 - * Exemplo: o lançamento de um MVP sem considerar dependências sistêmicas críticas levou a problemas em outras partes do sistema.
- Complexidade de Sistemas e Integrações: Testes em sistemas complexos e com múltiplas integrações foram particularmente desafiadores.
 - Altos Níveis de Conectores e Fluxos [9]:
 - * Projetos com centenas de conectores (mais de 300 em alguns casos) geraram dificuldades em cobrir todos os cenários possíveis, resultando em insucessos inesperados.

- * Testes frequentemente deixaram lacunas em cenários menos comuns, o que levou a falhas em funcionalidades críticas.
- Ambientes de Teste e Homologação [8]:
 - * Apesar do uso de ambientes dedicados, nem todos os problemas puderam ser antecipados devido à complexidade das combinações possíveis de conectores e configurações.
- Impactos Organizacionais e Mudanças de Equipe [11]: Fatores organizacionais também contribuíram para experiências de insucesso.
 - Corte de Posições Cruciais:
 - * Decisões de cortar custos e remover posições-chave, como analistas e agilistas, prejudicaram o fluxo de trabalho e levaram a dificuldades na evolução dos projetos.
 - Redistribuição de Responsabilidades:
 - * A falta de papéis bem definidos inicialmente gerou regressões no desenvolvimento, resultando em processos mais lentos e travados.
- Desafios com Automação de Testes [9 e 13]: A automação, apesar de ser uma ferramenta essencial, também foi apontada como fonte de insucessos em certos casos.
 - Falsos Negativos e Instabilidade: Scripts automatizados geraram falsos negativos recorrentes, dificultando a identificação de problemas reais.
 - Manutenção Excessiva: A automação tornou-se um trabalho pesado de manutenção, com esforços elevados para manter os testes existentes funcionando, limitando a inclusão de novos cenários.
 - Falhas de Ambiente: Problemas frequentes no ambiente de execução dificultaram análises rápidas e aumentaram o tempo de resolução de falhas.
- Lições Aprendidas e Impactos Psicológicos [10, 11 e 12]: Embora desafiadoras, as situações de insucesso trouxeram aprendizados importantes para os profissionais.
 - Aceitação como Parte do Processo: Apesar da sensação de frustração e improdutividade, os entrevistados reconheceram as falhas como oportunidades de aprendizado e melhoria.
 - Importância do Planejamento: Experiências destacaram a necessidade de uma preparação adequada e da colaboração entre equipes para mitigar riscos.
 - Adaptação e Resiliência: A recuperação gradual de processos após mudanças organizacionais demonstrou a capacidade de adaptação e aprendizado das equipes.

5.5.21 Comentários Adicionais sobre Qualidade, Métricas e Automação:

Os entrevistados compartilharam considerações adicionais que não haviam sido abordadas explicitamente nas perguntas anteriores. Esses comentários abrangem sugestões, reflexões e aspectos práticos.

- Feedback sobre a Entrevista [1, 3, 5, 9, 10 e 12]: Muitos entrevistados elogiaram o roteiro, destacando que as perguntas foram bem elaboradas e abordaram os principais tópicos.
 - Algumas respostas específicas:
 - * Parabenização pela entrevista: Apreciaram a abrangência das questões e a relevância do tema.
 - * Sem sugestões adicionais: Parte dos entrevistados considerou que não havia nada mais a ser incluído.
- Reflexões sobre o Papel do QA e a Cultura Organizacional [2 e 7]: Os entrevistados enfatizaram a importância do papel do QA e questões culturais no contexto de qualidade e automação.
 - Amplitude do Papel do QA: Sugeriram explorar mais o papel do QA em todas as etapas do ciclo de desenvolvimento, desde o planejamento até a entrega.
 - Futuro do QA: Refletiram sobre como o papel do QA está evoluindo, com foco crescente em automação e análise estratégica.
 - Cultura do "falhar e melhorar": Destacaram a necessidade de adotar uma mentalidade que valorize aprendizados obtidos por meio de falhas, promovendo melhorias contínuas.
- Aspectos Técnicos e Estratégicos sobre Automação [11 e 13]: A automação foi amplamente discutida, com ênfase em desafios e estratégias.
 - Desafios e Estratégias de Automação:
 - * Definir claramente o que, quando e por que automatizar foi destacado como um elemento crucial para o sucesso da automação.
 - * Custos, manutenção e resistência cultural foram citados como barreiras que precisam ser superadas.
 - Integração no CI/CD: A integração de automações no pipeline de desenvolvimento contínuo (CI/CD) foi mencionada como um objetivo essencial para maximizar a eficiência e reduzir riscos.

- Segurança e Desempenho do Sistema [8]: Outros aspectos técnicos relacionados à qualidade foram levantados.
 - Proteção contra Vulnerabilidades: A importância de investir em segurança foi destacada, com menção a um caso onde ataques cibernéticos prejudicaram a produtividade por semanas.
 - Escalabilidade e Desempenho: Garantir que o sistema mantenha desempenho sob aumento de carga ou volume foi considerado essencial.
- Ferramentas Complementares e Recursos de Aprendizado: Os entrevistados mencionaram ferramentas e práticas que enriquecem o trabalho com automação e métricas.
 - Ferramentas [4]:
 - * Postman, Appium, BrowserStack, LambdaTest, Azure DevOps, Jira e Zephyr foram destacadas como recursos importantes.
 - * O Jira, em especial, foi mencionado como uma ferramenta para obter métricas detalhadas, como criticidade de bugs e tipos de funcionalidades mais problemáticas.
 - Aprendizado e Colaboração [11]:
 - * Valorizar o conhecimento interno foi enfatizado como uma prática enriquecedora e eficiente. Interagir com colegas de projeto foi visto como mais prático do que buscar cursos externos.
 - * Além do aprendizado técnico, enfrentar medos e desafios relacionados à automação foi citado como um ganho pessoal significativo.

5.6 Discussão dos Resultados

Os resultados da pesquisa apresentaram insights interessantes que vão além das práticas convencionais. Esses elementos diferenciados oferecem uma visão rica sobre a evolução e os desafios enfrentados pelas equipes de tecnologia.

A Cultura de "Falhar e Melhorar" como Estratégia de Qualidade: A ênfase na cultura organizacional que valoriza o aprendizado a partir de falhas. Participantes destacaram a importância de reconhecer insucessos como parte natural do processo de desenvolvimento, incentivando melhorias contínuas. Essa abordagem aponta para uma cultura que fomenta experimentação, resiliência e evolução constante, alinhada aos princípios ágeis de adaptação rápida.

O Papel do QA no Ciclo de Vida do Desenvolvimento: Embora o papel do QA seja tradicionalmente associado à etapa de testes, alguns entrevistados ampliaram essa

visão, posicionando o QA como uma figura estratégica em todas as etapas do ciclo de desenvolvimento. Desde o planejamento inicial, passando pelo refinamento de histórias de usuário, até a entrega e monitoramento do produto, o QA é visto como um agente integrador, que conecta requisitos, desenvolvimento e qualidade.

Integração de Automação no CI/CD como Prioridade Estratégica: Embora a automação seja importante, a integração completa das automações no pipeline de CI/CD foi destacada como um diferencial estratégico. Essa prática permite feedbacks rápidos, monitoramento contínuo e redução de riscos durante o desenvolvimento.

Reflexões sobre Segurança e Escalabilidade: Os entrevistados enfatizaram aspectos técnicos como proteção contra vulnerabilidades cibernéticas e a importância de manter o desempenho do sistema sob aumento de carga. Um exemplo destacado foi o impacto de ataques cibernéticos em um projeto, que resultou em semanas de queda de produtividade. Essa perspectiva indica uma maturidade em relação à necessidade de qualidade não apenas funcional, mas também operacional, abordando desafios que nem sempre são tratados com prioridade em processos tradicionais.

Aprendizado Interno como Recurso Estratégico: Uma abordagem diferenciada foi o destaque à valorização do aprendizado interno em vez de depender exclusivamente de cursos externos. Profissionais relataram que interagir com colegas dentro do mesmo contexto de projeto não apenas economiza tempo, mas também enriquece o aprendizado de forma prática. Essa prática fortalece a colaboração e cria uma cultura de compartilhamento de conhecimento dentro das organizações.

Estratégias Híbridas de Metodologias: Embora Scrum e Kanban sejam amplamente utilizados, a combinação dessas metodologias (Scrumban) foi citada como uma prática inovadora e flexível. Essa abordagem permite ajustar o fluxo de trabalho conforme as demandas do projeto, mesclando planejamento estruturado e entregas contínuas.

5.7 Limitações

Este estudo, embora tenha apresentado resultados significativos, enfrentou algumas limitações que merecem destaque. Em primeiro lugar, o cronograma originalmente proposto sofreu impactos devido a problemas pessoais enfrentados pelo pesquisador. Esses desafios demandaram ajustes e priorização de atividades, o que influenciou diretamente na execução e no ritmo do trabalho. Outro ponto de dificuldade foi o processo de transcrição das entrevistas. Apesar do suporte tecnológico fornecido pela ferramenta de inteligência artificial VEED, que gerou transcrições iniciais, ajustes finos e revisões finais exigiram um tempo considerável. O esforço manual necessário para garantir a precisão dos dados revelou-se maior do que o esperado, tornando essa etapa operacionalmente desafiadora.

Nas análises textuais temáticas, também surgiram desafios relacionados ao refinamento dos temas. Embora os temas tenham sido identificados, a condensação e categorização conforme esperado pelo método foram mais trabalhosas do que o previsto, demandando iterações adicionais para garantir maior coerência e profundidade analítica. Por outro lado, um aspecto que inicialmente poderia ter sido considerado uma limitação, mas que se mostrou positivo, foi a busca pelos entrevistados. Contrariando as expectativas iniciais, o processo de recrutamento ocorreu de maneira fluida, com respostas favoráveis de profissionais dispostos a contribuir para o estudo, disponibilizando seu tempo e compartilhando experiências valiosas.

Finalmente, é importante reconhecer que a pesquisa não se encerra com este trabalho. Apenas o tempo alocado para sua execução chega ao fim. Os resultados apresentados são interessantes, mas há um potencial significativo para aprofundamento em futuras etapas. Novas rodadas de entrevistas e análises mais detalhadas poderiam enriquecer ainda mais as conclusões e ampliar a relevância do estudo.

5.8 Conclusões

O estudo revelou que as métricas são ferramentas indispensáveis para garantir a qualidade de software, permitindo monitorar o desempenho, identificar problemas e orientar decisões estratégicas. Contudo, a eficácia dessas métricas depende de sua contextualização e da maturidade dos processos de coleta e análise. O papel do QA está em transformação, ganhando uma abordagem mais estratégica e integrada ao ciclo de desenvolvimento, mas ainda enfrenta desafios como resistência cultural e falta de colaboração com outros stakeholders. A automação de testes foi amplamente reconhecida por sua capacidade de aumentar a eficiência, reduzir erros e melhorar a escalabilidade. No entanto, desafios como custos de manutenção, falsos negativos e instabilidades em ambientes de teste limitam seu uso pleno. Para maximizar os benefícios, é essencial investir em estratégias claras e personalizadas, priorizando cenários de alto impacto. Metodologias ágeis, como Scrum e Kanban, foram destacadas, muitas vezes adaptadas em combinações como Scrumban, que oferece flexibilidade e se ajusta a diferentes tipos de projetos. Além disso, questões como segurança cibernética, desempenho sob carga e resiliência do sistema emergiram como preocupações centrais, reforçando a necessidade de medidas proativas e estratégias robustas. Por fim, a valorização da cultura organizacional e do aprendizado contínuo demonstrou ser um diferencial. Promover uma mentalidade de “falhar e aprender”, incentivar a troca de conhecimento interno e integrar métricas e automação ao pipeline de desenvolvimento contínuo são práticas fundamentais para o sucesso na garantia da qualidade em software. Essas conclusões reforçam a importância de alinhar estratégias técnicas e culturais para superar os desafios e atender às demandas crescentes do mercado.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Baldin, N.; Munhoz, E. M. B. “Educação ambiental comunitária: uma experiência com a técnica de pesquisa snowball (bola de neve)”, *REMEA-Revista Eletrônica do Mestrado em Educação Ambiental*, vol. 27, 2011.
- [2] Caetano, C. “Melhores práticas na automação de testes”, *Revista Engenharia de Software Magazine*, vol. 5 Edição, 2008, pp. 42–47.
- [3] Crespo, A. N.; da Silva, O. J.; Borges, C. A.; Salviano, C. F.; de Teive, M.; Junior, A.; Jino, M. “Uma metodologia para teste de software no contexto da melhoria de processo”. In: Simpósio Brasileiro de Qualidade de Software (SBQS), 2004, pp. 204–2018.
- [4] ISO/IEC 25010. “ISO/IEC 25010:2011, systems and software engineering — systems and software quality requirements and evaluation — system and software quality models”, 2011.
- [5] Kunz, M.; Dumke, R. R.; Zenker, N. “Software metrics for agile software development”. In: Proc. of the Australian Conference on Software Engineering, 2008, pp. 673–678.
- [6] NETO, A. “Introdução a teste de software”, *Revista Engenharia de Software Magazine*, vol. 1ª Edição–Ano I, 2008, pp. 54–59.
- [7] Nicolette, D. “Introdução ao teste de software”. Elsevier Brasil, 2013, 408p.
- [8] Nicolette, D. “Software Development Metrics”. Manning, 2015, 192p.
- [9] Pressman, R.; Maxim, B. “Engenharia de software - 9.ed.” McGraw Hill Brasil, 2021, 704p.
- [10] Singh, G.; Singh, D.; Singh, V. “A study of software metrics”, *IJCEM International Journal of Computational Engineering & Management*, vol. 11–2011, 2011, pp. 22–47.
- [11] Sommerville, I. “Software engineering — 9th ed.” Pearson Education, 2011, 773p.

APÊNDICE A – PEDIDO DE CONSENTIMENTO

Os participantes foram consultados, durante o roteiro da entrevista, sobre a permissão da gravação da entrevista. Para fins acadêmicos, cujo apenas o áudio seria utilizado.

Pedido de Consentimento

(Realizado dentro do Roteiro da Entrevista)

"Gostaria de solicitar a sua autorização para gravar esta entrevista com propósito de poder transcrevê-la na íntegra para análise. A gravação, apenas do áudio, será utilizada exclusivamente para este propósito, com fins acadêmicos. A utilização da sua imagem bem como a sua identidade será mantida em sigilo. Qualquer informação que possa identificar você ou a empresa para a qual você trabalha será omitida para preservar a sua privacidade.

Você concorda em participar desta entrevista sob estas condições? Caso tenha alguma dúvida ou preocupação, estou à disposição para esclarecimentos."

APÊNDICE B – ROTEIRO DA ENTREVISTA

A pesquisa foi dividida em 4 seções de perguntas:

- 1 - Perfil do Entrevistado

Ordenação	Pergunta
1.1	Qual é a sua idade?
1.2	Com qual gênero você se identifica?
1.3	Qual é a sua formação acadêmica (nível e área)?
1.4	Qual é o seu cargo atual e quais são suas principais responsabilidades?
1.5	Em que tipo de empresa você trabalha (tamanho, setor)?
1.6	Em que tipo de projetos/produtos você vem trabalhando recentemente?

Tabela B.1 – Roteiro da Entrevista - Seção 1

- 2 - Contextualização / Conhecimento Geral

Ordenação	Pergunta
2.1	Quantos anos de experiência com Garantia da Qualidade de Produto? (Quanto tu te envolvereste, com que papéis e atividades esteve envolvido?)
2.2	Qual é a sua experiência com Automação de Teste?
2.3	Poderia descrever uma experiência marcante que você teve com a automação no desenvolvimento de software?
2.4	Numa escala de 0 a 10, quão familiarizado você se considera com os seguintes conceitos: de qualidade, de métricas e de automação de teste no desenvolvimento de software? Poderia explicar brevemente o que entende por cada um desses termos?
2.4.1	Quão familiarizado você se considera com o conceito de qualidade? Poderia explicar brevemente o que entende por qualidade?
2.4.2	Quão familiarizado você se considera com o conceito de métricas? Poderia explicar brevemente o que entende por métricas?
2.4.3	Quão familiarizado você se considera com o conceito de automação de testes? Poderia explicar brevemente o que entende por automação de testes?

Tabela B.2 – Roteiro da Entrevista - Seção 2

- 3 - Métricas de Qualidade

Ordenação	Pergunta
3.1	Quais as principais (ou mais importantes) métricas que estão sendo utilizadas para garantir a qualidade de um produto de software?
3.2	Como essas métricas são utilizadas no seu ambiente de trabalho? (Ex: Como são coletadas, quem usa, para que se usa)
3.3	Como essas métricas são definidas? (Quem define? Essas definições são revisitadas de tempos em tempos?)
3.4	Como você percebe o papel da automação de testes na coleta dessas métricas? O fato de existir automação de testes pode auxiliar na coleta das métricas? (Qual o contexto? Há automação ou ela é ausente? Poderia dar exemplos?)
3.5	Quais as lições aprendidas / benefícios você acredita que a automação de testes trouxe para a sua experiência em qualidade do produto?

Tabela B.3 – Roteiro da Entrevista - Seção 3

- 4 - Ferramentas e Metodologias Utilizadas

Ordenação	Pergunta
4.1	Quais ferramentas de execução ou de automação de testes você utiliza no desenvolvimento de software?
4.2	Quais metodologias de desenvolvimento de software são seguidas na sua organização?
4.3	Comente sobre experiências de insucesso de testes que ocorreram na tua experiência. Situações onde os testes deram errado, não necessariamente por culpa do QA, mas ainda assim ocorreu um insucesso?
4.4	Existe algum aspecto relacionado a qualidade, métricas ou automação que não foi abordado nas perguntas anteriores e que você gostaria de comentar?

Tabela B.4 – Roteiro da Entrevista - Seção 4

APÊNDICE C – DEBRIEFING QUESTIONS

Debriefing Questions

Local:

Date:

Interviewer:

Interviewee:

1. What were the key points observed about the focus?
(Quais foram os principais pontos observados durante a entrevista?)
 - a.

2. What did you find to be most surprising about this observation?
(O que mais te surpreendeu nessa entrevista?)
 - a.

3. What did you see or hear that was pretty much what you expected (or like other sites that you have seen)?
(O que foi mencionado que foi “mais do mesmo”?)
 - a.

4. What did you learn about the problem and “fixes” that you didn’t know before? That you did?
(O que foi aprendido sobre os problemas enfrentados e suas resoluções que não se sabia anteriormente?)
 - a.

5. What would you ask if we could go back? Would you ask the next participant this as well?
(Se fosse possível realizar a entrevista novamente, o que seria feito de diferente?)
 - a.

6. What worked really well?
(O que funcionou bem?)
 - a.

7. What didn’t work so well or what should be changed?
(O que não funcionou bem ou precisa ser modificado?)
 - a.

8. Other comments?

Figura C.1 – Debriefing Questions