



**Universidade Católica do Salvador
Bacharelado em Engenharia de Software**

**Adeilson Porto Moreira
Alisson Mendonça Melo
Amadeu Cerqueira de Almeida
Rodrigo Souza Silvestre
Tiago Galvão Pinho**

**Análise de Métricas de Software: A Influência das Métricas na
Qualidade de Software e na Melhoria dos Processos de
Desenvolvimento**

**Salvador
2025**

Adeilson Porto Moreira
Alisson Mendonça Melo
Amadeu Cerqueira de Almeida
Rodrigo Souza Silvestre
Tiago Galvão Pinho

Análise de Métricas de Software: A Influência das Métricas na Qualidade de Software e na Melhoria dos Processos de Desenvolvimento

Trabalho de Conclusão de Curso apresentado à Universidade Católica do Salvador como parte dos requisitos necessários para a obtenção do Título de Engenheiro de Software.
Orientador: Prof. Flavio Dusse

Universidade Católica do Salvador

Salvador
2025

Adeilson Porto Moreira
Alisson Mendonça Melo
Amadeu Cerqueira de Almeida
Rodrigo Souza Silvestre
Tiago Galvão Pinho

Análise de Métricas de Software: A Influência das Métricas na Qualidade de Software e na Melhoria dos Processos de Desenvolvimento

Trabalho de Conclusão de Curso apresentado à Universidade Católica do Salvador como requisito parcial para a obtenção do título de Engenheiro de Software.

Salvador, 30 de junho de 2025

Banca Examinadora:

Prof. Flavio Dusse
Universidade Católica do Salvador
Orientador

Prof. Glaucya Carreiro Boechat
Universidade Católica do Salvador

Prof. Cristiana Pereira Bispo
Universidade Católica do Salvador

Agradecimentos

Gostaríamos de agradecer ao professor Flavio Dusse, por todos os conselhos, pela ajuda e pela paciência com a qual guiou nosso aprendizado ao longo deste trabalho. Agradecemos a todos que participaram, direta ou indiretamente do desenvolvimento deste trabalho de pesquisa, e ajudaram a enriquecer o nosso trabalho.

Resumo

Este trabalho tem como objetivo analisar como diferentes categorias de métricas (produto, processo, projeto e recursos) são utilizadas por profissionais da área e qual o impacto dessa prática na eficiência das equipes e na qualidade dos produtos entregues. A pesquisa foi conduzida com uma abordagem exploratória e quantitativa, utilizando um questionário aplicado a 46 profissionais de desenvolvimento, qualidade e gestão de software. Os resultados revelam uma preferência pelo uso de métricas de processo e projeto, enquanto métricas de produto ainda são subaproveitadas, apesar de sua relevância técnica. Além disso, observou-se que a frequência e a forma como as métricas são revisadas variam significativamente entre os diferentes papéis nas equipes. Constatou-se que, quando bem aplicadas, as métricas deixam de ser meros indicadores e passam a desempenhar um papel ativo na tomada de decisões estratégicas, promovendo entregas mais eficientes, com menos retrabalho e maior alinhamento às necessidades do negócio. O estudo reforça a importância de uma cultura organizacional voltada à medição contínua, com foco na melhoria da qualidade e da maturidade dos processos de desenvolvimento.

Palavras-Chave: 1. Métricas de Software. 2. Qualidade de Software. 3. Testes. 4. Metodologias de Desenvolvimento.

Abstract

This project aims to analyze how different categories of metrics (product, process, project, and resources) are used by professionals in the field and what impact this practice has on team efficiency and the quality of delivered products. The research was conducted using an exploratory and quantitative approach, applying a questionnaire to 46 professionals. The results reveal a preference for using process and project metrics, while product metrics remain underutilized despite their technical relevance. Furthermore, it was observed that the frequency and manner in which metrics are reviewed vary significantly among the different roles within the teams. It was found that, when properly applied, metrics cease to be mere indicators and play an active role in strategic decision-making, promoting more efficient deliveries with less rework and greater alignment with business needs. The study reinforces the importance of an organizational culture focused on continuous measurement, with an emphasis on improving the quality and maturity of development processes.

Keywords: 1. Software Metrics. 2. Software Quality. 3. Testing. 4. Development Methodologies.

Lista de figuras

Figura 1 – Quadro na plataforma Trello utilizado para a gestão das atividades do projeto de pesquisa - Março de 2025	37
Figura 2 – Quadro na plataforma Trello utilizado para a gestão das atividades do projeto de pesquisa - Abril de 2025	38
Figura 3 – Quadro na plataforma Trello utilizado para a gestão das atividades do projeto de pesquisa - Maio de 2025	38
Figura 4 – Quadro na plataforma Trello utilizado para a gestão das atividades do projeto de pesquisa - Junho de 2025	38
Figura 5 – Distribuição de cargos dos participantes da pesquisa	40
Figura 6 – Utilização de ferramentas para coleta de métricas de software	41
Figura 7 – Metodologias de desenvolvimento mais utilizadas pelos participantes da pesquisa	42
Figura 8 – Frequência de utilização das categorias de métricas de software	44
Figura 9 – Frequência de utilização das categorias de métricas pelo perfil de Desenvolvedor/Engenheiro de Software	45
Figura 10 – Frequência de utilização das categorias de métricas pelo perfil de Gerente de Projetos	46
Figura 11 – Frequência de utilização das categorias de métricas pelo perfil de Analista de Qualidade	47
Figura 12 – Frequência de utilização das categorias de métricas pelo perfil de Scrum Master	48
Figura 13 – Frequência de revisão das métricas de qualidade	50
Figura 14 – Percepção dos profissionais sobre a contribuição das métricas na tomada de decisões estratégicas	51

Lista de tabelas

Tabela 1 – Modelo de Qualidade de Produto de Software segundo a ISO/IEC 25010 (ISO/IEC 25000 Users Group, 2024).	21
--	----

Lista de Siglas e Abreviaturas

ADS	Análise e Desenvolvimento de Sistemas
API	<i>Application Programming Interface</i>
BES	Bacharelado em Engenharia de Software
Ca/Ce	<i>Afferent Coupling / Efferent Coupling</i>
CI/CD	<i>Continuous Integration / Continuous Deployment (or Delivery)</i>
CMMI	<i>Capability Maturity Model Integration</i>
DevOps	<i>Development and Operations</i>
FTE	<i>Full Time Equivalent</i>
GQM	<i>Goal Question Metric</i>
IA	Inteligência Artificial
ISO/IEC	<i>International Organization for Standardization / International Electrotechnical Commission</i>
Istanbul	<i>Instrumental JavaScript Code Coverage Tool</i>
JaCoCo	<i>Java Code Coverage</i>
KLOC	<i>Kilo Lines of Code</i>
LCOM4	<i>Lack of Cohesion in Methods (version 4)</i>
MTBF	<i>Mean Time Between Failures</i>
MTTF	<i>Mean Time To Failure</i>
MTTR	<i>Mean Time To Repair</i>
PDCA	<i>Plan, Do, Check, Act</i>
PRIME	<i>Process Metrics</i>
PSP	<i>Process Personal Software</i>
QA	<i>Quality Assurance</i>
RUP	<i>Rational Unified Process</i>
SCADA	<i>Supervisory Control And Data Acquisition</i>
UCSAL	<i>Universidade Católica do Salvador</i>
UX	<i>User Experience</i>
XP	<i>Extreme Programming</i>

Sumário

1	INTRODUÇÃO	16
1.1	Objetivo Geral	17
1.2	Objetivos Específicos	18
1.3	Justificativa	18
1.4	Delimitação do Escopo	18
1.5	Assuntos a serem abordados	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	Qualidade de Software	20
2.1.1	Conceitos Fundamentais e a Importância da Qualidade	20
2.1.2	Dimensões da Qualidade de Software	21
2.1.3	Por que Gerenciar e Medir a Qualidade	22
2.2	Métricas de Software: Ferramentas para Quantificação	22
2.2.1	Definição, Propósitos e Benefícios	22
2.2.2	Classificação de Métricas de Software	23
2.2.2.1	Métricas de Produto	23
2.2.2.2	Métricas de Processo: Eficiência e Robustez	26
2.2.2.3	Métricas de Projeto: Prazo, Custo e Valor Percebido	27
2.2.2.4	Métricas de Recurso	29
2.3	Processos de Desenvolvimento de Software	31
2.3.1	Modelos de Ciclo de Vida	32
2.3.2	O Conceito de Melhoria Contínua de Processos	32
2.3.3	Profissionais Determinantes para o Desenvolvimento de Software	32
2.4	Fechamento do Referencial Teórico	33
3	METODOLOGIA	35
3.1	Classificação da Pesquisa	35
3.1.1	Coleta de dados	35
3.1.2	Local de Estudo	36
3.1.3	Apresentação dos Resultados	36
3.1.4	Aspectos Éticos da Pesquisa	36
3.2	Gestão do projeto	37
3.2.1	Evolução do projeto	37
4	EXPERIMENTO E COLETA DE DADOS	39
4.1	Introdução à Pesquisa de Campo	39

4.2	Distribuição de Perfil dos Participantes	39
4.3	Ferramentas e Metodologias Utilizadas	41
4.3.1	Uso de Ferramentas para Coleta de Métricas	41
4.3.2	Metodologias de Desenvolvimento Predominantes	42
4.4	Categorias de Métricas mais Adotadas	43
4.4.1	Cargo X Frequência de Utilização de Métricas de Acordo com a Categoria	44
4.4.1.1	Perspectiva do Desenvolvedor/Engenheiro de Software	45
4.4.1.2	Perspectiva do Gerente de Projetos	46
4.4.1.3	Perspectiva do Analista de Qualidade/QA	47
4.4.1.4	Perspectiva do Scrum Master	48
4.4.1.5	Análise geral	49
4.5	Frequência de Revisão das Métricas	49
4.6	Contribuição das Métricas na Tomada de Decisões	50
4.7	Análise dos Resultados	52
4.7.1	Perfil dos Profissionais e Contexto Metodológico	52
4.7.2	Práticas de Medição: Ferramentas e Frequência	52
4.7.3	Utilização e Priorização das Categorias de Métricas	53
4.7.4	Perspectivas Diferentes por Cargo Profissional	53
4.7.5	Impacto Estratégico das Métricas	53
5	CONSIDERAÇÕES FINAIS	55
5.1	Retomada da Pesquisa e Síntese dos Resultados	55
5.2	Resposta aos Objetivos da Pesquisa	56
5.3	Implicações do Estudo	57
5.4	Limitações da Pesquisa	58
5.5	Sugestões para Trabalhos Futuros	59
5.6	Conclusão	59
	REFERÊNCIAS	61
A	RESULTADOS DA PESQUISA	65
B	INSTRUMENTO DE COLETA DE DADOS	66

1 Introdução

No cenário tecnológico atual, o uso de softwares permeia virtualmente todas as esferas da atividade humana, desde operações empresariais críticas até interações sociais cotidianas. O crescente uso e dependência de sistemas de software cada vez mais complexos impõe uma demanda igualmente crescente por produtos que não apenas funcionem, mas que o façam com **qualidade** (ÁVILA et al., 2024). Garantir a qualidade de um software transcende a mera ausência de erros; envolve também a capacidade do produto de atender consistentemente às necessidades dos seus usuários e *stakeholders*, agregando valor real a eles (SOUZA, 2016). A falha em alcançar níveis adequados de qualidade pode levar não apenas à insatisfação do cliente e do usuário, mas também a prejuízos financeiros e operacionais significativos (SOUZA, 2016), tornando a qualidade um fator estratégico fundamental.

Garantir que um software tenha qualidade não é uma tarefa simples. Isso porque o conceito de qualidade vai muito além de apenas “funcionar bem”, ele envolve uma série de características importantes, como funcionalidade, confiabilidade, usabilidade, eficiência, facilidade de manutenção, portabilidade, entre outras. Essas dimensões são bem descritas em modelos como o da ISO/IEC 25010 (SOUZA, 2016). O grande desafio está em avaliar essas características de forma objetiva e mantê-las ao longo de todo o ciclo de desenvolvimento. Como algumas dessas qualidades podem ser interpretadas de maneiras diferentes por cada pessoa envolvida no processo e os sistemas de hoje são cada vez mais complexos, é fundamental adotar métodos mais rigorosos e mensuráveis para gerir e garantir a qualidade do software de forma eficaz.

Nesse contexto, as **métricas de software** emergem como ferramentas indispensáveis para proporcionar uma base quantitativa na avaliação da qualidade (SOUZA, 2016). Elas permitem medir diversas características específicas de um produto de software (como complexidade do código, coesão, acoplamento) e do processo de seu desenvolvimento (como esforço, tempo, produtividade) (MILLS, 1988). Ao transformar atributos muitas vezes abstratos em dados numéricos, as métricas oferecem a possibilidade de monitorar o estado do projeto, identificar potenciais problemas precocemente e auxiliar na tomada de decisões mais embasadas em evidências (SOUZA, 2016; RAMOS, 2022).

Apesar de possuírem um grande potencial, a aplicação efetiva dessas métricas na indústria ainda acaba enfrentando barreiras significativas (SOUZA, 2016; LEAL, 2023). Simplesmente coletar dados não é o suficiente; o verdadeiro desafio reside em selecionar as métricas adequadas de acordo com os objetivos específicos do projeto e do seu contexto organizacional (LEAL, 2023) e interpretar corretamente seus valores é

frequentemente dificultado pela ausência de valores de referência (*thresholds*) bem estabelecidos e validados (SOUZA, 2016). A dificuldade na seleção e avaliação de métricas apropriadas para contextos específicos, especialmente os ágeis, torna complexa a implementação de um processo de medição eficaz que realmente impulse a melhoria contínua (LEAL, 2023). A falta de ligação clara entre a medição e ações concretas de melhoria no processo limita o potencial transformador das métricas.

Diante desse panorama, o objetivo central desta monografia é analisar métricas de qualidade de software, investigando a sua influência na melhoria dos processos de desenvolvimento. Pretende-se explorar como a análise sistemática de métricas pode transcender uma simples avaliação da qualidade do produto final e servir como um mecanismo efetivo para identificar gargalos, orientar decisões e promover o aprimoramento contínuo das práticas e fluxos de trabalho no desenvolvimento de software.

A relevância desta pesquisa reside na necessidade urgente das empresas em otimizar seus processos para entregar produtos de maior qualidade de forma mais eficiente e previsível. Compreender a influência das métricas nos processos de desenvolvimento pode fornecer *insights* valiosos para que equipes e gestores utilizem a medição não apenas como um diagnóstico, mas como uma ferramenta proativa de gestão e evolução (SOUZA, 2016). Ao investigar a relação entre métricas e processos, este trabalho busca preencher lacunas na compreensão de como essa influência ocorre na prática, consolidando conhecimentos dispersos na literatura e oferecendo uma visão integrada sobre o tema. A análise aprofundada das métricas e sua conexão com a melhoria de processos pode fornecer subsídios importantes para equipes de desenvolvimento, gerentes de projeto e pesquisadores da área.

1.1 Objetivo Geral

Este trabalho tem como objetivo geral analisar a influência das métricas de software na qualidade de desenvolvimento de projetos de software. A proposta é investigar como sua aplicação sistemática, ou não, contribui para a eficiência nos processos de desenvolvimento e para o aprimoramento da qualidade do produto final.

Os resultados desta investigação e sua aplicação possuem o potencial de tornar os processos de desenvolvimento mais transparentes e controlados, otimizando a alocação de recursos, reduzindo o número de defeitos e gastos com retrabalho e aprimorando a capacidade das equipes de entregar softwares de alta qualidade de maneira mais rápida e consistente.

1.2 Objetivos Específicos

- Identificar e analisar as principais métricas de software relevantes para a avaliação e o aprimoramento dos processos de desenvolvimento, com base na literatura e em uma pesquisa de campo;
- Realizar uma pesquisa por meio de um formulário online distribuído para profissionais da área e discutir os resultados obtidos;
- Selecionar as métricas consideradas mais relevantes para a avaliação da qualidade do software e o aprimoramento dos processos de desenvolvimento, com base na literatura e nas respostas levantadas no formulário, observando o impacto das métricas selecionadas na qualidade do software e na eficiência do processo de desenvolvimento;
- Investigar como os *insights* derivados da análise das métricas podem ser utilizados para direcionar ações que visem melhorar a qualidade do software e otimizar os processos de desenvolvimento, mapeando os principais desafios e boas práticas de implementação e uso de métricas com foco na melhoria contínua dos processos;

1.3 Justificativa

O desenvolvimento deste trabalho se justifica pela necessidade de investigar mais a fundo qual a real influência que as métricas de software possuem na melhoria dos processos de desenvolvimento, pois é fundamental compreender não apenas quais métricas podem ser usadas, mas como sua análise pode efetivamente direcionar intervenções no processo, identificar gargalos, validar a eficácia de novas práticas e contribuir para um ciclo de desenvolvimento mais eficiente e voltado à **garantia de qualidade do software**.

Ao focar em analisar essa influência das métricas, esta pesquisa busca contribuir tanto para o corpo de conhecimento acadêmico quanto para a indústria, oferecendo *insights* que podem auxiliar equipes e gestores a utilizarem as métricas de forma mais estratégica e impactante.

1.4 Delimitação do Escopo

O presente estudo será composto pela análise de um conjunto de métricas selecionadas a partir da literatura e de um formulário respondido por profissionais do mercado, critérios para seleção de métricas e orientações para facilitar a seleção e utilização das métricas.

Não será desenvolvido nenhum software para auxiliar na seleção de métricas, também não serão criadas novas métricas além das já existentes na literatura e das coletadas do mercado.

1.5 Assuntos a serem abordados

A fim de alcançar os objetivos propostos, este trabalho foi estruturado em cinco capítulos. O presente Capítulo 1, **Introdução**, foi exposta uma visão geral do objetivo e motivação desta pesquisa. O Capítulo 2, **Fundamentação Teórica**, estabelece as bases conceituais da pesquisa, abordando os temas de qualidade de software, a classificação e aplicação de métricas, e os processos de desenvolvimento. Em seguida, o Capítulo 3, **Metodologia**, detalha os procedimentos adotados na investigação, incluindo a classificação da pesquisa, o método de coleta de dados e as técnicas utilizadas para a análise dos resultados. O Capítulo 4, **Experimento e Coleta de Dados**, dedica-se à apresentação e discussão aprofundada dos resultados obtidos na pesquisa de campo, explorando os padrões de uso de métricas e as percepções dos profissionais da área. Por fim, o Capítulo 5, **Considerações Finais**, sintetiza os principais achados do estudo, responde aos objetivos de pesquisa, discute as implicações práticas e teóricas dos resultados e aponta direções para possíveis trabalhos futuros.

2 Fundamentação Teórica

Este capítulo estabelece as bases conceituais necessárias para a compreensão da análise de métricas de software e sua influência na melhoria dos processos de desenvolvimento. Serão abordados os conceitos fundamentais de qualidade de software, métricas de software e os modelos de processos de desenvolvimento.

2.1 Qualidade de Software

A busca pela qualidade é inerente à engenharia em suas diversas disciplinas e na Engenharia de Software não é diferente. Contudo, definir e mensurar a qualidade de um produto intangível como o software apresenta desafios particulares. A qualidade de software não é um atributo monolítico, mas sim um conceito multifacetado que pode ser percebido de diferentes formas dependendo da perspectiva em que for observado (usuário, desenvolvedor, gerente, etc.).

2.1.1 Conceitos Fundamentais e a Importância da Qualidade

As métricas de qualidade de software são instrumentos fundamentais para transformar atributos abstratos e intangíveis do produto e do processo de software em parâmetros quantitativos e mensuráveis. Essa quantificação possibilita um controle rigoroso da qualidade, fundamentando decisões técnicas e gerenciais ao longo de todo o ciclo de vida do software.

Conforme estabelecido nas normas ISO/IEC 25010 e ISO/IEC 25040, as métricas se organizam em categorias que abrangem diferentes aspectos do desenvolvimento e da operação do software: métricas de produto, processo, projeto e desempenho. Cada categoria tem um papel específico na avaliação da qualidade, produtividade e valor do software, sendo subdividida em subtipos que detalham atributos particulares.

A ISO/IEC 25010 define oito características principais de qualidade: funcionalidade, confiabilidade, capacidade de interação (substituindo usabilidade), eficiência, segurança (com reforço significativo), compatibilidade, manutenibilidade e flexibilidade (anteriormente chamada de portabilidade). Cada uma dessas características é subdividida em sub características avaliáveis por métricas específicas, permitindo uma avaliação detalhada e alinhada às necessidades do produto e do usuário (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2023).

Para facilitar a compreensão do modelo de qualidade de produto proposto pela norma ISO/IEC 25010, foi elaborada uma tabela-síntese (Tabela 1). A referida tabela foi construída a partir da tradução e adaptação de um organograma visual da norma

(ISO/IEC, 2023), consolidando as características e subcaracterísticas de qualidade em um formato de tabela para análise.

Tabela 1 – Modelo de Qualidade de Produto de Software segundo a ISO/IEC 25010 (ISO/IEC 25000 Users Group, 2024).

QUALIDADE DE PRODUTO DE SOFTWARE									
ADEQUAÇÃO FUNCIONAL	EFICIÊNCIA DE DESEMPENHO	COMPATIBILIDADE	USABILIDADE	CONFIABILIDADE	SEGURANÇA	MANUTENIBILIDADE	FLEXIBILIDADE	SEGURANÇA OPERACIONAL	
Completude funcional Correção funcional Apropriação funcional	Comportamento em relação ao tempo	Coexistência	Reconhecimento de adequação	Maturidade	Confidencialidade	Modularidade	Adaptabilidade	Restrição operacional	
			Apreensibilidade						Disponibilidade
	Utilização de recursos	Interoperabilidade	Proteção contra erros do usuário	Tolerância a falhas	Não repúdio	Rastreabilidade	Instalabilidade	Prova de falhas	
			Engajamento do usuário						Recuperabilidade
	Capacidade		Inclusividade		Resistência				Integração segura
			Assistência ao usuário						
			Autodescritividade						

O processo de avaliação, conforme a ISO/IEC 25040, envolve a especificação de requisitos, planejamento e execução de testes, análise dos resultados e tomada de decisão, aplicável tanto a produtos em desenvolvimento quanto finalizados. A escolha adequada das métricas é fundamental para garantir que as avaliações sejam relevantes, acionáveis e capazes de apoiar a melhoria contínua da qualidade (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2011).

A evolução do conceito de qualidade de software acompanha a própria maturidade da engenharia de software. Enquanto nas décadas de 1970 e 1980 a preocupação era majoritariamente com a ausência de defeitos, atualmente a qualidade é vista como um conjunto de atributos que inclui segurança, experiência do usuário, interoperabilidade e sustentabilidade (FENTON; BIEMAN, 2014).

Além disso, a qualidade é reconhecida como fator estratégico para o sucesso de produtos e serviços digitais, impactando diretamente a competitividade organizacional (HUMPHREY, 1996).

2.1.2 Dimensões da Qualidade de Software

As métricas de produto avaliam atributos intrínsecos do software, como complexidade, tamanho e defeitos, fornecendo dados objetivos sobre confiabilidade, estabilidade e desempenho. Métricas de processo medem a eficiência e robustez das atividades de desenvolvimento e testes, enquanto métricas de projeto avaliam o desempenho em termos de prazo, custo e satisfação do cliente. Por fim, as métricas de desempenho e confiabilidade monitoram o comportamento do software em operação, garantindo que os requisitos não funcionais críticos sejam atendidos.

A ISO/IEC 25010 divide a qualidade em três perspectivas:

- Qualidade interna (características do código e arquitetura, como coesão e acoplamento);
- Qualidade externa (comportamento do software em execução, como desempenho e robustez);
- Qualidade em uso (experiência do usuário, incluindo efetividade, produtividade, segurança e satisfação). A relação entre essas dimensões é fundamental: melhorias em atributos internos, como manutenibilidade, impactam positivamente características externas e a experiência do usuário (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2016).

2.1.3 Por que Gerenciar e Medir a Qualidade

A percepção subjetiva de qualidade é insuficiente devido à complexidade dos sistemas e à multiplicidade de stakeholders. Fenton & Bieman (2014) (FENTON; BIEMAN, 2014) destacam que métricas objetivas são essenciais para avaliar, controlar e melhorar a qualidade de maneira sistemática.

Métricas confiáveis devem ser objetivas, válidas, sensíveis e normalizadas, permitindo decisões baseadas em dados. Estudos nacionais (SERPRO, 2023) mostram que a ausência de métricas leva à subestimação de riscos e aumento de falhas em produção.

2.2 Métricas de Software: Ferramentas para Quantificação

As métricas de software são ferramentas essenciais na engenharia de software, servindo como uma base quantitativa para a avaliação de diversos aspectos tanto de produto de software quanto do processo de seu desenvolvimento. Elas são capazes de transformar características muitas vezes abstratas -como complexidade, qualidade ou nível de esforço- em dados numéricos mensuráveis e comparáveis. Essa quantificação é crucial para monitorar o progresso de projetos, identificar potenciais problemas de maneira precoce, auxiliar no controle da qualidade e facilitar a tomada de decisões mais embasadas em evidências ao longo de todo o ciclo de vida de um software.

2.2.1 Definição, Propósitos e Benefícios

Métricas de software são funções que transformam atributos do processo ou produto em valores quantitativos. Elas auxiliam na compreensão, avaliação, controle e previsão da qualidade do software (KANER; FALK; NGUYEN, 1999).

A integração de métricas em ambientes ágeis e DevOps acelera *feedbacks* e permite ajustes rápidos.

Metodologias ágeis são abordagens de desenvolvimento de software que promovem a colaboração contínua entre equipes e clientes, dividindo o trabalho em ciclos curtos e iterativos para entregar valor de forma incremental, facilitando a adaptação rápida a mudanças e o controle transparente do progresso (CONCEIÇÃO, 2024; HERNANDEZ; GOMES et al., 2023). Já o DevOps é uma cultura e conjunto de práticas que unem desenvolvimento (Dev) e operações (Ops), automatizando processos de integração, testes e entrega contínua, o que reduz o tempo entre a criação de uma funcionalidade e sua disponibilização em produção, aumentando a qualidade e a confiabilidade do software.

2.2.2 Classificação de Métricas de Software

Usualmente, as métricas são divididas entre: métricas de produto, que descrevem as características do software em si (como tamanho, complexidade ou número de defeitos); métricas de processo, que avaliam a eficácia e eficiência do ciclo de vida do desenvolvimento (como tempo de desenvolvimento ou nível de experiência da equipe); métricas de recursos, que monitoram a alocação e utilização de ativos, como as horas de trabalho da equipe; métricas de projeto, que mensuram aspectos como prazo, custo e valor percebido, essenciais para o controle e sucesso do projeto. Essa categorização auxilia na seleção de indicadores apropriados para diferentes objetivos de análise e melhoria.

2.2.2.1 Métricas de Produto

As métricas de produto são fundamentais para o controle da qualidade do software em teste e em produção. Elas fornecem dados objetivos sobre a confiabilidade, estabilidade e desempenho do sistema, baseando-se fortemente na avaliação dos defeitos encontrados durante os testes. Tais defeitos, que podem ser simples contagens ou estatísticas complexas, indicam a necessidade de mudanças no produto, pois refletem que os requisitos ainda não foram plenamente satisfeitos.

A análise dos defeitos permite estimar a confiabilidade atual do software e prever sua evolução conforme os testes avançam e os defeitos são corrigidos. Além disso, métricas como complexidade ciclomática, linhas de código, coesão e acoplamento ajudam a identificar áreas de risco e a necessidade de refatoração, facilitando a manutenção e a evolução do sistema.

- **Complexidade Ciclométrica**

“Ao trabalhar com métricas de código, um dos itens menos compreendidos parece ser a complexidade ciclomática. Essencialmente, com a complexidade ciclomática, números mais altos são ruins e números mais baixos são bons. Você pode usar a complexidade ciclomática para ter uma ideia de quão difícil determinado código pode ser para testar, manter ou depurar, além de servir como um indicativo da probabilidade de o código gerar erros. Em um nível mais geral, determina-se o valor da complexidade ciclomática contando o número de decisões feitas no seu código-fonte”

(Microsoft Corporation, 2024)

Proposta por McCabe (MCCABE, 1976), mensura o número de caminhos independentes em um grafo de fluxo de controle, dado por:

$$v(G) = E - N + 2P$$

- **E** é o número de arestas,
- **N** o número de nós,
- **P** o número de componentes conectados.

“O número exato a ser usado como limite, no entanto, continua sendo um tanto controverso. O limite original de 10, proposto por McCabe, possui evidências significativas que o sustentam, mas limites de até 15 também têm sido usados com sucesso. Limites acima de 10 devem ser reservados para projetos que possuem várias vantagens operacionais em relação aos projetos típicos, como por exemplo, equipe experiente, design formal, uma linguagem de programação moderna, programação estruturada, revisões de código e um plano de testes abrangente. Em outras palavras, uma organização pode adotar um limite de complexidade maior que 10, mas apenas se tiver certeza do que está fazendo e estiver disposta a dedicar o esforço adicional de testes exigido por módulos mais complexos.”

(MCCABE, 1976)

Valores acima de 20 indicam necessidade de refatoração e maior rigor nos testes, pois refletem lógica excessivamente ramificada.

- **Caso de uso**

Segundo relatório técnico da NASA (*National Aeronautics and Space Administration*), “o padrão atualizado para *Software Assurance and Software Safety* contém um requisito para avaliação da complexidade ciclomática em todos os módulos de código críticos para a segurança, estabelecendo o limite máximo em 15” (NASA Engineering

and Safety Center, 2020). Ferramentas como SonarQube desenvolvida pela empresa SonarSource automatizam essa medição, emitindo alertas para módulos que ultrapassam limites recomendados (SonarSource, 2006).

- **Linhas de Código (LOC)**

Linhas de Código (LOC) é um indicador bruto do volume de software produzido, não conta linhas em branco e comentadas; embora não reflita qualidade, sua evolução serve como proxy de produtividade bruta (GAMBACORTA et al., 2024).

Indicador bruto do volume de software, conta linhas não em branco e não comentadas. Embora LOC isoladamente não seja suficiente para avaliar qualidade, sua correlação com defeitos por KLOC (mil linhas de código) e *churn* (volume de alterações) pode indicar produtividade e complexidade (TRODOD, 2009).

Exemplo prático: No desenvolvimento do kernel Linux, áreas com alta densidade de defeitos por KLOC são identificadas para revisão e testes mais rigorosos, auxiliando na priorização de esforços de manutenção.

- **Coesão (LCOM4) e Acoplamento (Ca/Ce)**

- LCOM4: Mede a coesão interna de classes em sistemas orientados a objetos. Valores maiores que 1 sugerem baixa coesão e necessidade de refatoração para separar responsabilidades.
- Acoplamento:
 - Ca** (Aferente): Número de classes que dependem de uma classe;
 - Ce** (Eferente): Número de classes das quais uma classe depende.

Baixos valores de acoplamento facilitam manutenção e extensibilidade.

Aplicação: LCOM4 pode ser aplicada para avaliar classes em sistemas orientados a objetos, como em projetos Java, em que a baixa coesão indica a necessidade de separar responsabilidades para facilitar a manutenção (SILVA et al., 2024).

- **Taxa de Defeitos (Defeitos/KLOC) e Code Churn**

A densidade de defeitos, calculada como número de erros por mil linhas de código (KLOC), oferece uma avaliação direta da qualidade do produto. Valores elevados de densidade de defeitos sinalizam a urgência de reforço nos processos de testes e revisão de requisitos. (TRODOD, 2009).

- Defeitos/KLOC: Indica qualidade do produto; valores elevados requerem reforço de testes e revisão de requisitos.

- *Code Churn*: Mede o volume de alterações no código ao longo do tempo, sinalizando instabilidade ou dificuldades no entendimento das funcionalidades.

Caso real: Em projetos ágeis, o monitoramento do code churn ajuda a identificar áreas instáveis do código que frequentemente mudam, sinalizando risco de introdução de novos defeitos. Por exemplo, times do setor financeiro monitoram *churn* para priorizar testes e revisões em áreas críticas.

- **Cobertura de Código por Testes**

Percentual de código exercitado por testes automatizados, medido por ferramentas como JaCoCo (Java) e Istanbul (JavaScript). Alta cobertura reduz risco de regressões, mas não substitui a qualidade dos testes (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2023).

Uso em um pipeline CI/CD: Builds com cobertura abaixo de 80% podem ser bloqueados para evitar regressões, garantindo maior confiabilidade do software entregue.

2.2.2.2 Métricas de Processo: Eficiência e Robustez

As métricas de processo avaliam a eficiência e a robustez das práticas de desenvolvimento, incluindo o tempo de ciclo, frequência de *commits*, taxa de falha em *deploys* e cobertura de testes, fornecendo indicadores para otimização contínua dos fluxos de trabalho.

- **Lead Time e Cycle Time**

Lead Time e *Cycle Time* são métricas cruciais em metodologias DevOps (Desenvolvimento e Operações), permitindo que empresas monitorem o tempo entre o início do desenvolvimento e a entrega em produção, otimizando a resposta a mudanças de mercado (KANTORSKI; KROTH, 2018).

Lead Time Intervalo do início do desenvolvimento até a entrega em produção;

Cycle Time Tempo do início da codificação até o deploy.

Caso de uso: Empresas como Netflix e Amazon monitoram essas métricas para acelerar entregas e garantir rápida resposta a mudanças, otimizando o fluxo de trabalho DevOps.

- **Taxa de Falha de Alteração (*Change Failure Rate*)**

A Taxa de Falha de Alteração é um indicador crucial da estabilidade dos deploys em produção. Uma alta taxa de falhas pode indicar a necessidade de revisão dos processos de teste e validação (ROSENBERG; HAMMER; SHAW, 1998).

Percentual de deploys que resultam em falhas em produção, exigindo rollback ou hotfix.

Importância: Taxas abaixo de 15% indicam maturidade em testes e validação. Times de software crítico, como em saúde, usam essa métrica para garantir estabilidade.

- **Cobertura de Testes e Taxa de Aprovação**

A Cobertura de Testes, combinada com a Taxa de Aprovação, oferece uma visão abrangente da eficácia dos testes, permitindo que as equipes de QA ajustem suas estratégias para garantir a qualidade do software (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2023).

- **Tempo Médio de Resolução de Defeitos (MTTR)**

O Tempo Médio de Resolução de Defeitos (MTTR) é um indicador crítico para a manutenção ágil, especialmente em sistemas de missão crítica, em que um baixo MTTR é essencial para minimizar o impacto de falhas (ALVES et al., 2016).

Relevância: Em telecomunicações, MTTR baixo é essencial para cumprir SLA e garantir alta disponibilidade.

- **Frequência de Commits**

A Frequência de *Commits*, combinada com a prática de *code review* integrado, contribui para a redução de conflitos e o aumento da qualidade do código, promovendo um ritmo sustentável de desenvolvimento (KANTORSKI; KROTH, 2018).

Prática recomendada: Volume de *commits* por desenvolvedor ou equipe, *proxy* de atividade e fluxo contínuo. Alta frequência de commits com *code reviews* integrados reduz riscos de integração e melhora qualidade. Projetos open source como Kubernetes adotam essa prática.

2.2.2.3 Métricas de Projeto: Prazo, Custo e Valor Percebido

A mensuração de desempenho em projetos complexos utiliza indicadores como Variação de Prazo (SV) e Variação de Custo (CV), permitindo detectar desvios orçamentários e cronológicos em tempo real (MEDEIROS, 2023). A satisfação do cliente, medida por CSAT e NPS, é crucial para aferir a aderência ao mercado e guiar melhorias (ALVES et al., 2016), enquanto o Retorno Sobre o Investimento (ROI) justifica investimentos e compara alternativas. Além disso, a Produtividade da Equipe, influenciada por funcionalidades entregues, commits e lead time, suporta decisões de alocação e identificação de impedimentos, sendo todos esses indicadores essenciais para uma gestão eficaz e alinhada aos objetivos de negócio.

• Variação de Prazo e Custo (SV e CV)

Dados de 127 projetos de infraestrutura analisados pela UFU revelaram que equipes que monitoram essas métricas semanalmente reduziram estouros orçamentários em 37% comparado às que usavam métodos tradicionais (MEDEIROS, 2023).

- Variação de Prazo (SV): $SV=VA-VP$;
- Variação de Custo (VC): $VC=VA-CA$.

Onde VA = Valor Agregado, VP = Valor Planejado, CA = Custo Atual.

Aplicação: A implementação desses indicadores exige integração entre sistemas de gestão e fluxos de trabalho. O estudo destaca casos em que a automatização da coleta de dados aumentou a precisão das previsões em 52%, particularmente em projetos com fases interdependentes (MEDEIROS, 2023).

• Satisfação do Cliente (CSAT e NPS)

A satisfação do cliente, avaliada por meio de pesquisas CSAT e NPS, é um indicador crucial da lealdade do cliente e da qualidade do produto (ALVES et al., 2016).

- CSAT: Avalia satisfação pontual (escala 1–5);
- NPS: Mede intenção de recomendação (–100 a +100).

Uso: Empresas SaaS aplicam NPS para medir lealdade e orientar melhorias, correlacionando altos índices a menor *churn*.

• Produtividade da Equipe

A Produtividade da Equipe, medida por meio de funcionalidades entregues e *lead time*, auxilia na identificação de gargalos e no balanceamento da carga de trabalho, otimizando o processo de desenvolvimento (KANTORSKI; KROTH, 2018).

Indicadores como funcionalidades entregues, *commits*, *churn* e *lead time* suportam decisões de alocação e identificação de impedimentos.

• Retorno Sobre o Investimento (ROI)

A análise do Retorno sobre Investimento (ROI) em projetos de tecnologia requer a comparação entre custos de desenvolvimento e ganhos operacionais tangíveis. O estudo da Universidade Federal de Uberlândia (UFU) estabelece a fórmula abaixo:

$$ROI = \frac{\text{Benefícios} - \text{Custos}}{\text{Custos}} \quad (2.1)$$

Essa fórmula permite quantificar o impacto financeiro de iniciativas de melhoria contínua (LIMA, 2017; NEVES, 2022). Em projetos analisados entre 2020-2024, observou-se que cada R\$1,00 investido em ferramentas de automação gerou retornos médios de R\$2,30 em redução de retrabalho (NEVES, 2022). O mesmo estudo alerta para armadilhas comuns na avaliação de ROI, como a subestimação de custos indiretos (ex.: treinamento de equipes) e a supervalorização de ganhos de curto prazo (LIMA, 2017). Para mitigar esses riscos, recomenda-se a adoção de horizontes temporais mínimos de 18 meses para avaliação de resultados (NEVES, 2022). O Retorno Sobre o Investimento (ROI) é fundamental para justificar investimentos em qualidade e novas funcionalidades, comparando custos de desenvolvimento e ganhos financeiros ou operacionais (LIMA, 2017).

2.2.2.4 Métricas de Recurso

As métricas de recurso são essenciais para monitorar e otimizar a alocação e utilização de ativos humanos em projetos de desenvolvimento de software. Elas fornecem dados objetivos sobre a eficiência operacional, capacidade da equipe e utilização de recursos, permitindo aos gestores tomar decisões fundamentadas sobre alocação de pessoal, planejamento de capacidade e otimização de processos (PINHEIRO et al., 2023; SOARES; ZAIDAN; JAMIL, 2015). A análise de recursos permite estimar a capacidade atual da equipe e prever a evolução da produtividade conforme o projeto avança. Métricas como horas trabalhadas, utilização de recursos, velocidade da equipe e throughput ajudam a identificar gargalos operacionais e a necessidade de rebalanceamento de cargas de trabalho, facilitando o planejamento estratégico e a gestão eficiente de projetos (SHAH et al., 2023; MRIDUL; RANA, 2016).

• **Equivalente de Tempo Integral (FTE - Full Time Equivalent)**

O FTE é uma métrica fundamental que quantifica a carga de trabalho de um funcionário em relação a um funcionário em tempo integral. Essa métrica padroniza a medição de recursos humanos independentemente do regime de trabalho individual.

Fórmula de cálculo:

$$\text{FTE} = (\text{Horas Efetivamente Trabalhadas}) / (\text{Horas Padrão de Trabalho})$$

Onde:

- Horas Efetivamente Trabalhadas: tempo dedicado a atividades produtivas;
- Horas Padrão de Trabalho: carga horária completa (tipicamente 40 horas/semana)

Estudos demonstram que o monitoramento de FTE permite redução de até 50% no tempo gasto em processos manuais e melhora significativa na alocação de recursos (RAHMAN; ANDRESWARI; FAUZI, 2022). A implementação adequada de controle de FTE pode resultar em economia substancial, medida por meio da otimização da utilização de recursos humanos. Caso de uso: Em projetos de desenvolvimento ágil, o FTE é utilizado para planejar a alocação de consultores em múltiplos projetos simultâneos. Por meio do monitoramento semanal do FTE, a gestão consegue identificar rapidamente profissionais subutilizados e realocá-los para projetos com maior demanda, evitando custos de ociosidade e melhorando a eficiência operacional. Ferramentas como o Microsoft Power BI ¹ são empregadas para automatizar a visualização dos dados de FTE, permitindo ajustes dinâmicos na alocação de recursos.

• Utilização de Recursos (*Resource Utilization*)

A utilização de recursos mede o percentual de tempo produtivo em relação ao tempo total disponível de cada membro da equipe. Essa métrica é crucial para identificar ociosidade ou sobrecarga de trabalho (BONA, 2022).

Fórmula de cálculo:

$$\text{Utilização (\%)} = (\text{Horas Produtivas} / \text{Horas Disponíveis}) \times 100$$

Valores ideais situam-se entre 70-85%, permitindo margem para atividades não planejadas e manutenção. Utilização acima de 90% pode indicar risco de *burnout*, enquanto valores abaixo de 60% sugerem subutilização de recursos (BONA, 2022). Aplicação prática: Sistemas de monitoramento em tempo real demonstram que equipes com utilização otimizada apresentam 86,8% de redução no tempo de espera por atendimento em ambientes multi-projeto (FISCHER et al., 2024). A implementação de algoritmos de elasticidade proativa e reativa permite ajustar dinamicamente a alocação de recursos conforme a demanda (FISCHER et al., 2024).

• *Throughput* e Produtividade

O *throughput* quantifica o volume de entregáveis concluídos por unidade de tempo, fornecendo visibilidade sobre a capacidade produtiva da equipe (SHAH et al., 2023; SYNOVIC et al., 2022).

Métricas de *throughput*:

- Itens completados por sprint: número de user stories/tarefas finalizadas;
- Lead time: tempo total desde solicitação até entrega;
- Cycle time: tempo de desenvolvimento ativo;
- Takt time: intervalo entre conclusões de itens consecutivos.

¹ <https://www.microsoft.com/pt-br/power-platform/products/power-bi>

A produtividade em desenvolvimento de software correlaciona-se com múltiplos fatores, incluindo comunicação da equipe. Ferramentas de monitoramento como a plataforma PRIME (*Process Metrics*) oferecem métricas longitudinais incluindo produtividade, densidade de issues e fator de risco (*bus factor*), proporcionando insights sobre processo além de produto (SYNOVIC et al., 2022).

• Capacidade e Planejamento de Recursos

O planejamento de capacidade projeta necessidades futuras de recursos baseado em demanda esperada e capacidade histórica da equipe (PINHEIRO et al., 2023; VALLEJO et al., 2024).

Componentes do planejamento:

- Capacidade nominal: recursos teóricos disponíveis
- Capacidade efetiva: recursos reais considerando fatores de ajuste;
- Fatores de contingência: reservas para atividades imprevistas;
- Projeções de demanda: estimativas de trabalho futuro.

Estudos com *Process Personal Software* (PSP) demonstram que métricas de produtividade melhoram significativamente a disciplina estudantil e permitem seguimento mais preciso do progresso. Registros incluem linhas de código por hora, tempo de desenvolvimento por programa e número de erros (VALLEJO et al., 2024).

Exemplo de implementação: Sistemas de alocação computacional baseados em arquitetura orientada a serviços podem reduzir o tempo gasto por gestores na elaboração de cronogramas e controlar custos e prazos em projetos com restrições de recursos. Tais sistemas integram-se a ferramentas como Microsoft Project², Primavera³ e OpenProject⁴ para otimização automática.

2.3 Processos de Desenvolvimento de Software

O processo de desenvolvimento de um software é um conjunto de atividades estruturadas para criar e manter sistemas, desde a ideia inicial à manutenção. Esses processos visam gerenciar o trabalho, garantir qualidade e gerir riscos. Modelos como Cascata ou Ágil (Scrum, Kanban, etc.) oferecem diferentes fluxos, adaptáveis ao contexto de cada projeto e a adesão a um processo bem definido é crucial para a previsibilidade e o sucesso de um projeto de software, impactando diretamente nos resultados obtidos.

² <https://www.microsoft.com/pt-br/microsoft-365/project/project-management>

³ <https://www.oracle.com/br/construction-engineering/primavera-p6/>

⁴ <https://www.openproject.org/>

2.3.1 Modelos de Ciclo de Vida

Os modelos de ciclo de vida de software, como Cascata, Scrum, Kanban e DevOps, orientam as etapas de desenvolvimento e manutenção.

- Cascata: Métricas de prazo e custo em milestones;
- Scrum: Velocity e burndown charts em sprints;
- DevOps: Lead time, change failure rate e frequência de deploys.

Estudos recentes demonstram que a implementação de métricas modernas, como as DORA Metrics (DevOps Research and Assessment) e o SPACE Framework, está associada a melhorias significativas na eficiência operacional de projetos de software de grande escala. Pesquisas publicadas pela Codecon (2025) (??) destacam reduções de 30-45% no Tempo Médio de Recuperação (MTTR) e aumentos de 20% na frequência de deploys em equipes brasileiras que adotaram esses modelos. O framework SPACE, por sua vez, permite integrar métricas técnicas (como ciclo de desenvolvimento) com indicadores organizacionais (como satisfação de equipes), criando uma visão holística da produtividade.

2.3.2 O Conceito de Melhoria Contínua de Processos

A melhoria contínua é sustentada pelo ciclo PDCA (*Plan-Do-Check-Act*), na qual a medição sistemática permite identificar oportunidades de aprimoramento, testar hipóteses e padronizar práticas bem-sucedidas (HUMPHREY, 1996)(Humphrey, 1996).

Modelos de maturidade como o CMMI reforçam a importância da medição e da análise de dados para evolução dos processos, com organizações maduras apresentando maior previsibilidade, menor retrabalho e maior satisfação do cliente (CMMI Institute, 2025).

2.3.3 Profissionais Determinantes para o Desenvolvimento de Software

O desenvolvimento de software é, por essência, uma atividade que reúne múltiplas disciplinas e depende da colaboração integrada entre diferentes perfis profissionais ao longo de todo o ciclo de vida do produto. Cada função tem um papel fundamental para garantir que o sistema desenvolvido vá além dos requisitos técnicos, atendendo também às demandas de negócio, usabilidade, segurança e manutenção contínua.

Entre os papéis mais recorrentes em equipes de desenvolvimento, destacam-se os profissionais de engenharia de requisitos, desenvolvedores, analistas de testes e qualidade, gerentes de projeto e designers de experiência do usuário. Cada um contribui

com uma visão específica sobre o produto e os processos, promovendo decisões mais completas e eficazes. Essa diversidade de perfis permite à equipe atuar de forma multidimensional, considerando tanto aspectos técnicos quanto estratégicos na entrega do software.

Com a adoção crescente de metodologias ágeis e práticas de integração e entrega contínua, novas funções têm se consolidado nos times modernos, como engenheiros de DevOps, especialistas em segurança, analistas de dados e Scrum Masters. Esses profissionais desempenham papéis importantes na automação, no monitoramento e na garantia da escalabilidade das soluções. Como destacam Pressman e Maxim (PRESSMAN; MAXIM, 2016), a evolução das práticas de engenharia de software exige que as equipes sejam compostas por especialistas capazes de colaborar em ciclos cada vez mais rápidos e iterativos.

Apesar das responsabilidades específicas de cada função, o sucesso de um projeto depende da atuação conjunta e harmônica entre todos os membros da equipe. Uma comunicação clara, a definição de responsabilidades bem estabelecidas e o alinhamento contínuo entre os objetivos técnicos e organizacionais são elementos essenciais para a construção de um ambiente produtivo e eficiente. Segundo Sommerville (SOMMERVILLE, 2011), a colaboração bem estruturada entre os diferentes papéis é um dos principais fatores que influenciam a qualidade e a previsibilidade na entrega de software.

Além disso, criar uma cultura de equipe baseada em confiança, feedback e propósito compartilhado é um fator determinante para o desempenho organizacional. De acordo com Lencioni (LENCIONI, 2012), equipes disfuncionais não fracassam por falta de conhecimento técnico, mas pela ausência de coesão e comprometimento coletivo com os resultados.

Reconhecer a importância dos diferentes profissionais envolvidos no desenvolvimento de software é, portanto, essencial para fortalecer a cultura de qualidade. Uma equipe bem estruturada, capacitada e integrada tende a gerar produtos mais robustos, sustentáveis e alinhados às necessidades do mercado.

2.4 Fechamento do Referencial Teórico

Ao longo deste capítulo, foram estabelecidos os pilares conceituais que sustentam a presente pesquisa, partindo da natureza multifacetada da qualidade de software, passando pelas ferramentas para sua quantificação – as métricas de software – e culminando na importância dos processos de desenvolvimento como o contexto no qual a qualidade é construída e aprimorada.

Ficou evidente que a qualidade está longe de ser um atributo único e sim um conjunto de características (como funcionalidade, confiabilidade, usabilidade e manute-

nibilidade), conforme modelado por normas como a ISO/IEC 25010. A pesquisa demonstrou que a gestão eficaz dessas características depende intrinsecamente da capacidade de medi-las de forma objetiva.

Nesse sentido, as métricas de software foram apresentadas como o instrumento fundamental para essa quantificação. A classificação das métricas em categorias – Produto, Processo, Projeto e Recursos – permitiu compreender como cada uma oferece uma perspectiva distinta, porém complementar, para a avaliação do ciclo de vida do software. Seja analisando a complexidade do código (Produto), a eficiência do fluxo de trabalho (Processo) ou o controle de custos e prazos (Projeto), as métricas fornecem os dados necessários para transformar a percepção subjetiva de qualidade em *insights* acionáveis.

Por fim, o capítulo conectou esses conceitos ao demonstrar que a aplicação e a interpretação das métricas estão diretamente ligadas ao processo de desenvolvimento adotado, seja ele um modelo ágil focado em ciclos curtos ou uma abordagem DevOps orientada ao fluxo contínuo. A melhoria contínua da qualidade, sustentada pelo ciclo PDCA, emerge como o objetivo final, em que a medição sistemática alimenta a tomada de decisões e impulsiona a evolução dos processos.

3 Metodologia

Para este capítulo serão apresentados os procedimentos da metodologia utilizada para a execução desta pesquisa. Muito se passa pela forma como o projeto foi gerenciado ao longo do desenvolvimento, no qual será detalhado a classificação do estudo, os métodos de coleta e análise de dados, o local em que a pesquisa foi realizada e os cuidados éticos envolvidos.

A metodologia de pesquisa é o conjunto de técnicas e métodos usado pelos pesquisadores para guiar durante a jornada de construção de seus trabalhos, na qual se obtém, analisa e interpreta dados de forma sistemática e objetiva (GIL, 2008) salienta que a metodologia é essencial para assegurar a validade e a confiabilidade dos resultados, pois rege o percurso da pesquisa desde a definição do problema até a apresentação das conclusões. Toda escolha de uma boa metodologia vai se basear no tipo do problema estudado e dos objetivos que desejam alcançar.

3.1 Classificação da Pesquisa

Com base nas definições expostas sobre o que de fato é uma metodologia bem construída. Esta pesquisa pode ser classificada como exploratória quanto aos seus objetivos, pois tem por finalidade compreender como métricas de qualidade de software são utilizadas na prática e como podem contribuir para melhoria dos processos de desenvolvimento. Essa classificação se justifica pela intenção de investigar um tema ainda em expansão no meio acadêmico e profissional, instigando percepções, práticas e o conhecimento no nível de utilização de métricas de qualidade de software pelos usuários.

A abordagem do problema se caracteriza como quantitativa pois permite analisar estatísticas dos dados coletados, visando identificar padrões e tendências entre os participantes da pesquisa, feita por meio da coleta de dados via questionários e todo seu estudo sendo exposto e trabalhado via gráficos e tabelas.

Quanto a natureza da pesquisa, é caracterizada como aplicada pois tem como objetivo adquirir novos conhecimentos para o desenvolvimento ou aprimoramento de produtos, processos e sistemas.

3.1.1 Coleta de dados

A coleta de dados foi realizada por meio da aplicação de um questionário estruturado, elaborado com base na revisão da literatura sobre métricas de qualidade de software. O questionário continha perguntas fechadas as quais visavam entender e

avaliar o conhecimento do participante por meio de suas experiências profissionais, sendo ele um gestor ou não. Por fim, também foi adicionada uma pergunta de cunho aberto para que o participante tivesse a oportunidade de mostrar ferramentas ou processo de sua utilização diária.

O questionário foi disponibilizado online, utilizando uma plataforma de formulários digitais¹, o que possibilitou o alcance de profissionais e estudantes da área de desenvolvimento de software em diferentes localidades.

3.1.2 Local de Estudo

Este estudo foi realizado em ambiente acadêmico tendo como foco da pesquisa os profissionais da área de Tecnologia da Informação e sua vivência em desenvolvimento de software. Para isso, foi elaborado um questionário com perguntas direcionadas a essa temática.

O questionário foi divulgado em grupos de redes sociais e ambientes virtuais relacionados ao desenvolvimento de software, contando também com a participação de alunos e professores dos cursos de Graduação em Bacharelado em Engenharia de Software (BES) e em Análise e Desenvolvimento de Sistemas (ADS) da Universidade Católica do Salvador (UCSAL).

3.1.3 Apresentação dos Resultados

Os dados foram processados e analisados com o auxílio de ferramentas de planilhas do Google. Utilizou-se também a plataforma Google Looker Studio², que possibilitou a geração de gráficos, tabelas e análises comparativas.

Os resultados foram apresentados com base no referencial teórico da pesquisa. Isso permitiu identificar como as métricas de qualidade são compreendidas e aplicadas na prática, bem como seus efeitos nos processos de desenvolvimento de software.

3.1.4 Aspectos Éticos da Pesquisa

A pesquisa respeitou todos os princípios éticos aplicáveis a estudos com seres humanos. A participação foi voluntária e os participantes foram informados sobre os objetivos da pesquisa.

Garantiu-se o anonimato e a confidencialidade das respostas. Nenhum dado pessoal identificável foi coletado e os resultados foram utilizados exclusivamente para fins acadêmicos.

¹ <https://docs.google.com/forms>

² <https://lookerstudio.google.com/>

3.2 Gestão do projeto

A gestão de todo processo da pesquisa foi realizada de forma colaborativa entre os cinco participantes do grupo, com reuniões semanais para o acompanhamento do progresso, discussão de ideias e definição de tarefas. O trabalho foi dividido de forma justa, respeitando as habilidades e experiências dos participantes, promovendo assim uma atuação mais dinâmica e eficiente.

O grupo contou com a orientação ativa do professor Flávio Dusse, que participou das reuniões, forneceu feedbacks contínuos e orientações metodológicas. Todo o percurso foi registrado e compartilhado via ferramenta Trello³ e grupos fechados de rede social, garantindo o controle do cronograma e a organização das etapas do projeto.

3.2.1 Evolução do projeto

Para o planejamento das atividades, utilizou-se um quadro na ferramenta Trello, estruturado com base na metodologia ágil Kanban. O quadro foi organizado em colunas intituladas "A Fazer", "Em Andamento" e "Concluído", além de colunas adicionais como "Links Importantes" e "Referências", que auxiliaram na organização dos materiais de apoio e informações relevantes para o desenvolvimento do projeto. As tarefas foram categorizadas e distribuídas entre os membros da equipe de acordo com a afinidade e competência de cada um em relação às atividades propostas.

Exposto abaixo está a evolução dos quadros ao longo do projeto mês a mês (Figuras 1 a 4):

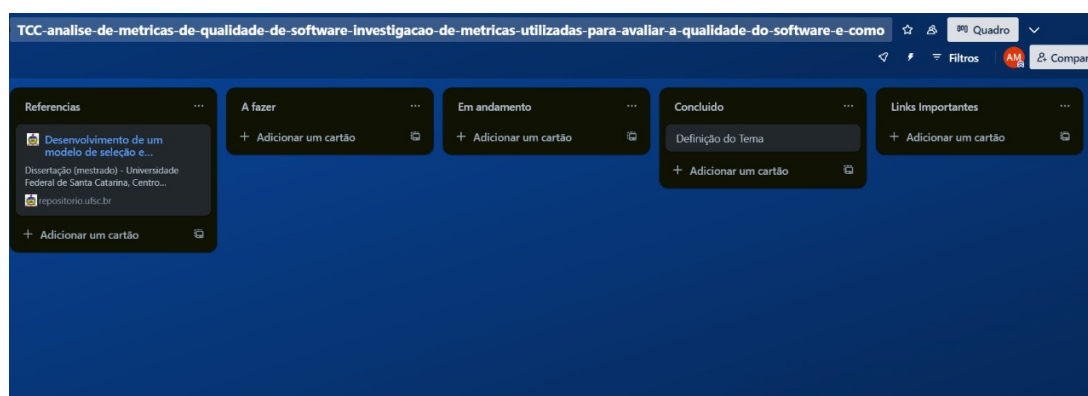


Figura 1 – Quadro na plataforma Trello utilizado para a gestão das atividades do projeto de pesquisa - Março de 2025

³ <https://trello.com>

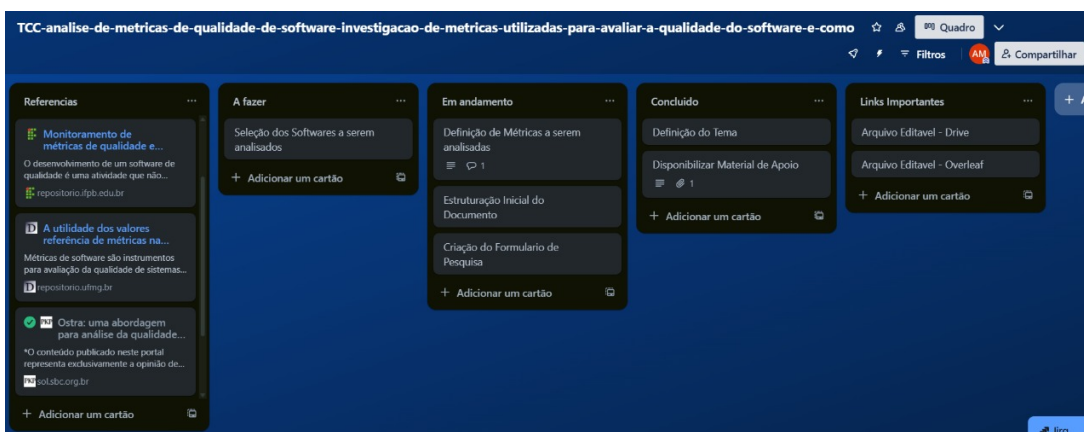


Figura 2 – Quadro na plataforma Trello utilizado para a gestão das atividades do projeto de pesquisa - Abril de 2025

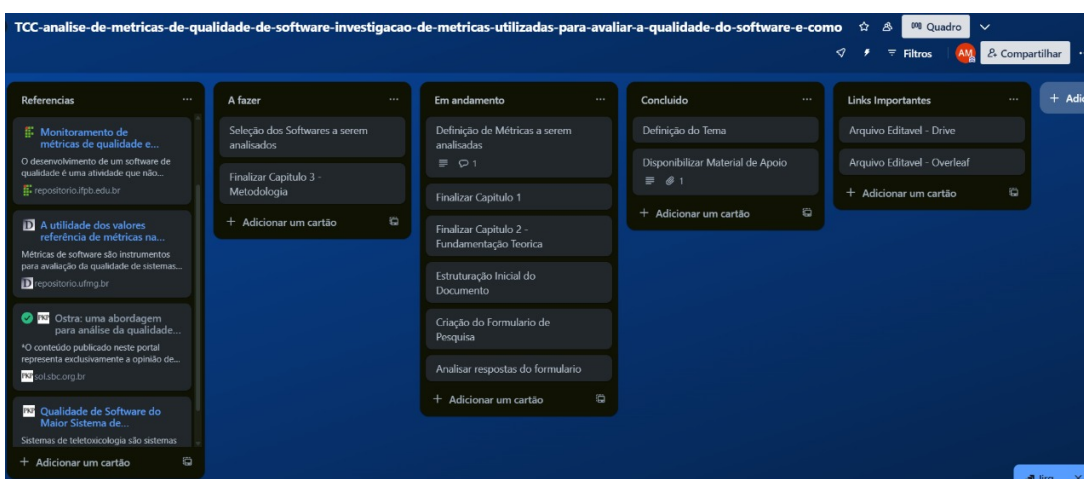


Figura 3 – Quadro na plataforma Trello utilizado para a gestão das atividades do projeto de pesquisa - Maio de 2025



Figura 4 – Quadro na plataforma Trello utilizado para a gestão das atividades do projeto de pesquisa - Junho de 2025

4 Experimento e Coleta de Dados

4.1 Introdução à Pesquisa de Campo

Para entender como as métricas de qualidade de software são utilizadas e monitoradas em ambientes de desenvolvimento reais, foi distribuído um questionário direcionado a profissionais da área de desenvolvimento, qualidade e gestão de projetos de software. O objetivo principal desta pesquisa, foi investigar como o uso das métricas de software pode influenciar na eficiência dos processos de desenvolvimento e no aprimoramento da qualidade de produtos de software. Enquanto a literatura nos fornece uma base teórica com estudos consolidados, a pesquisa de campo busca trazer uma visão pragmática, refletindo os desafios, as ferramentas e as métricas efetivamente empregadas no dia a dia das equipes de desenvolvimento.

A aplicação do questionário foi realizada de maneira remota, tendo sido distribuído em grupos de redes sociais e ambientes virtuais relacionados ao desenvolvimento de software, buscando respostas de profissionais atuantes de diferentes contextos da área de desenvolvimento de software. Foram obtidas 46 respostas válidas, que formam a base para a análise apresentada a seguir. O questionário abordou temas como as metodologias de desenvolvimento utilizadas, o emprego de ferramentas para coleta de métricas, as métricas específicas adotadas, a percepção sobre a importância e os desafios do uso de métricas e a influência observada na melhoria dos processos.

4.2 Distribuição de Perfil dos Participantes

Para contextualizar os dados que serão abordados ao longo deste capítulo, é fundamental compreender o perfil dos profissionais que participaram do estudo. A pesquisa alcançou uma amostra diversificada de 46 profissionais, cujos cargos refletem as múltiplas facetas do desenvolvimento de software. A distribuição dos participantes, conforme detalhado na Figura 5 abaixo, demonstra uma representação significativa dos principais papéis envolvidos na criação e gestão de produtos de software.

A análise do gráfico revela que a maioria dos respondentes se concentra em três áreas centrais:

- Desenvolvedores/Engenheiros de Software (26,1%): Representando a maior fatia, esse grupo traz a perspectiva técnica da construção do software, estando diretamente envolvida com a qualidade interna do código e a implementação de funcionalidades;

- Gerentes de Projeto (19,6%): O segundo maior grupo é composto por gestores, responsáveis pelo planejamento, execução e controle dos projetos. Suas respostas tendem a refletir uma preocupação com prazos, custos e a entrega de valor, conectando as métricas aos objetivos de negócio;
- Analistas de Qualidade/QA (15,2%): Esse grupo expressivo de profissionais oferece uma visão focada na validação e verificação do produto, garantindo que os requisitos sejam atendidos e que o software funcione conforme o esperado.

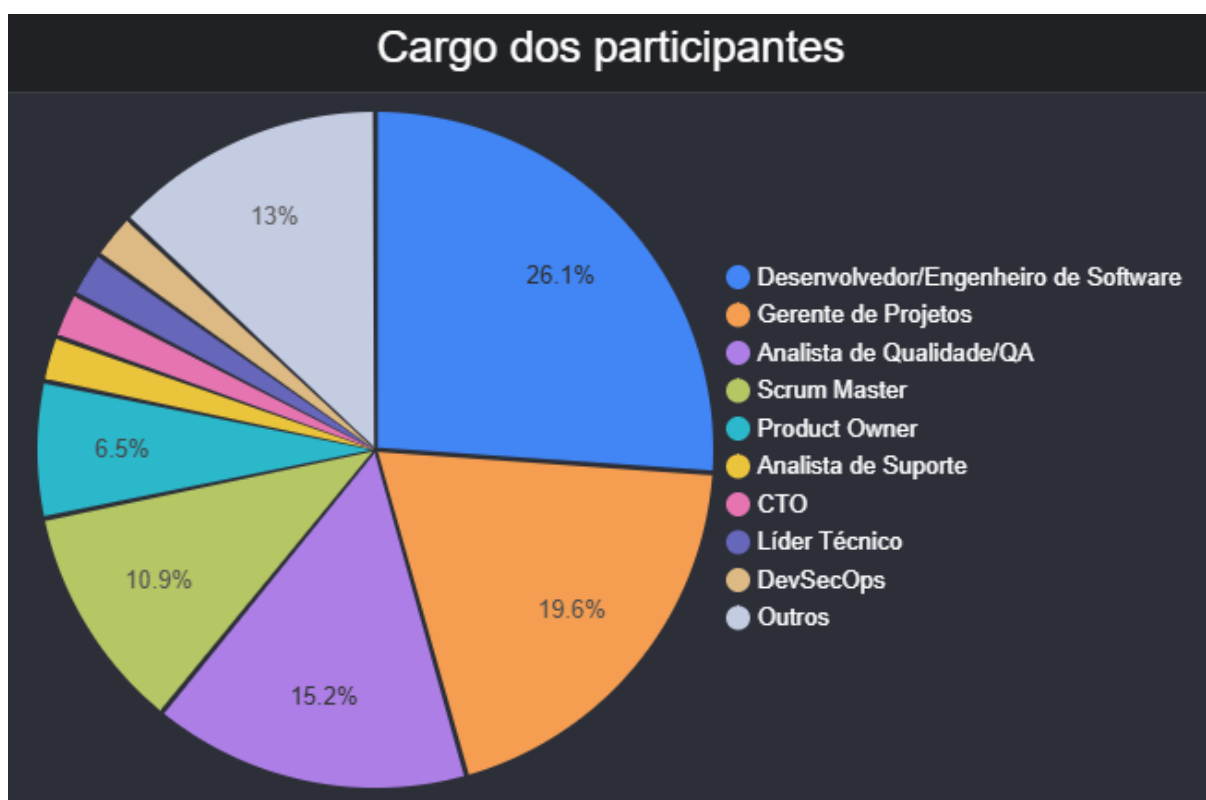


Figura 5 – Distribuição de cargos dos participantes da pesquisa

Além desses três papéis, a pesquisa também contou com a participação de outros papéis cruciais em ambientes ágeis, como Scrum Masters (10,9%) e Product Owners (6,5%), que contribuem com a perspectiva do processo e do valor do produto, respectivamente. A presença de Líderes Técnicos, CTOs, Analistas de Suporte e DevSecOps também enriquece ainda mais a amostra.

Essa diversidade de papéis permite analisar o tema sob diferentes óticas, contrastando as preocupações e prioridades de quem desenvolve, quem gerencia e quem testa. Os *insights* derivados dessa amostra heterogênea fornecem uma visão abrangente e representativa dos desafios e práticas mais utilizadas na indústria de software.

4.3 Ferramentas e Metodologias Utilizadas

Um primeiro aspecto investigado na pesquisa, foi o uso de ferramentas e as abordagens metodológicas empregadas pelas equipes dos respondentes.

4.3.1 Uso de Ferramentas para Coleta de Métricas

Questionados sobre o uso de ferramentas de análise estática e/ou de gestão de projetos para obter ou coletar métricas de qualidade, os resultados indicam uma adoção significativa, mas não universal.

Um primeiro aspecto investigado na pesquisa, foi o uso de ferramentas e as abordagens metodológicas empregadas pelas equipes dos respondentes exposto em forma de gráfico na Figura 6 abaixo.

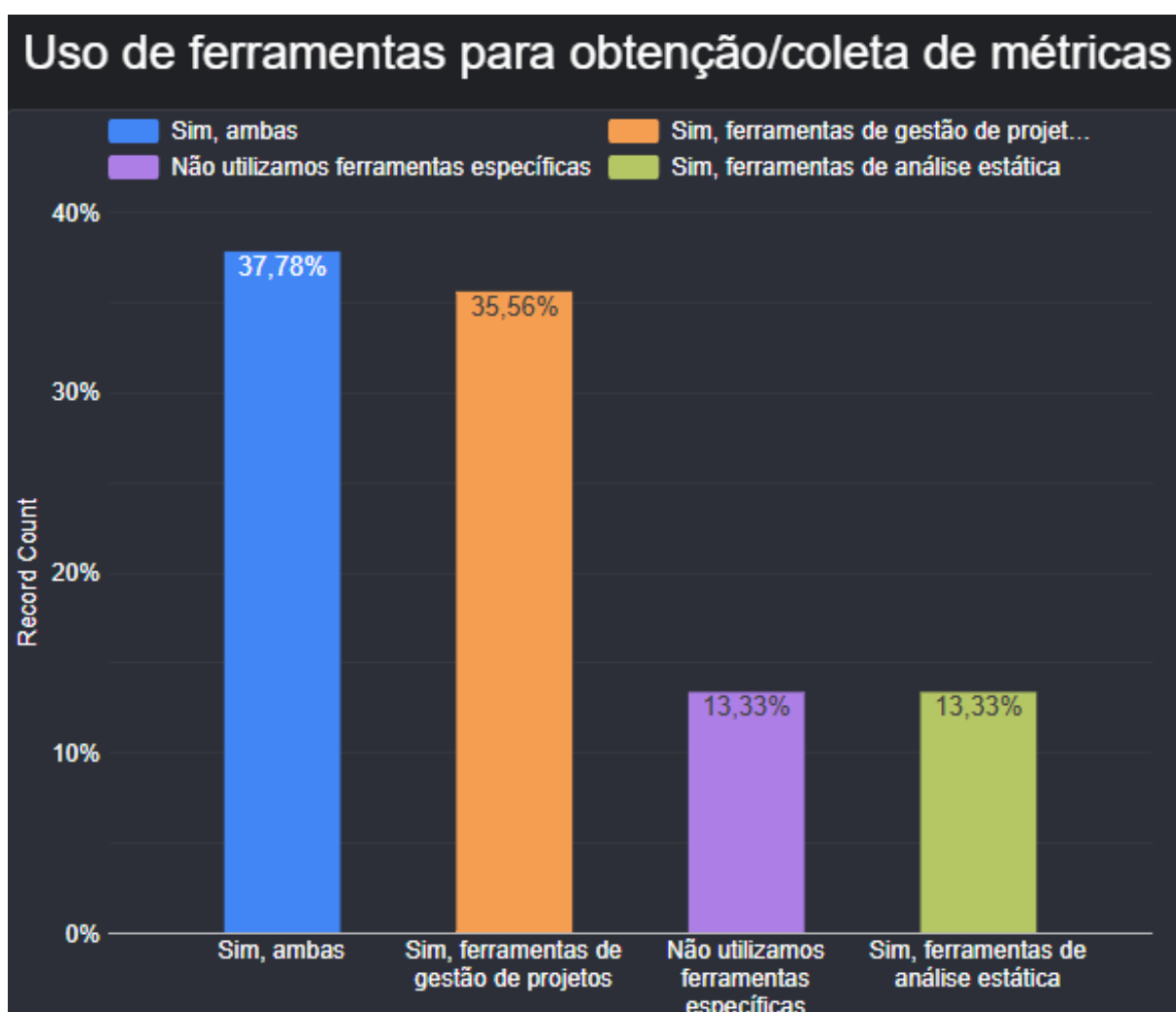


Figura 6 – Utilização de ferramentas para coleta de métricas de software

- 37,8% dos participantes afirmaram utilizar ambas as categorias de ferramentas (análise estática e gestão de projetos);

- 35,6% dos participantes relataram usar apenas ferramentas de gestão de projetos;
- 13,3% dos participantes indicaram usar apenas ferramentas de análise estática;
- 13,3% dos participantes responderam não utilizar ferramentas específicas para esse fim.

Estes dados sugerem que, embora a maioria das equipes (cerca de 85% dos que responderam afirmativamente ou negativamente) utilize algum tipo de ferramenta para apoiar a coleta de métricas, há uma ligeira predominância do uso de ferramentas de gestão de projetos em detrimento das de análise estática, quando usadas isoladamente. A combinação de ambas, no entanto, representa a maior parcela individual, alinhando-se com a visão da literatura sobre a importância da automação (RAMOS, 2022; ÁVILA et al., 2024) e da integração de dados de diferentes fontes. A existência de equipes que não utilizam ferramentas específicas (13,3%) aponta para uma oportunidade ou desafio na adoção de práticas mais sistematizadas de medição.

4.3.2 Metodologias de Desenvolvimento Predominantes

Quanto às metodologias de desenvolvimento, as respostas abertas indicaram uma forte predominância de abordagens ágeis.

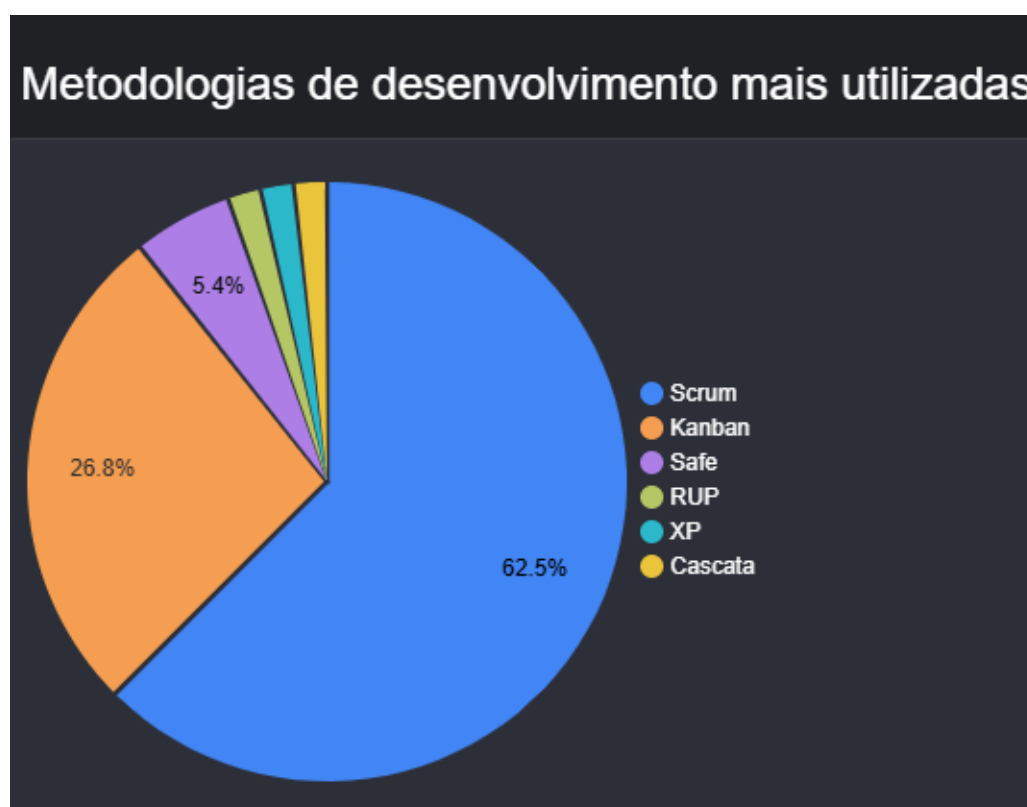


Figura 7 – Metodologias de desenvolvimento mais utilizadas pelos participantes da pesquisa

Os resultados da pesquisa expostos no gráfico da Figura 7 acima, demonstram claramente a prevalência do Scrum como metodologia dominante, sendo mencionado por 35 respondentes (62,5% da amostra). O Kanban aparece em segundo lugar, com 15 menções (26,8%), frequentemente utilizado em conjunto com o Scrum. Frameworks de escala como o SAFe (Scaled Agile Framework) foram citados por 3 respondentes (5,4%), enquanto o modelo Cascata, XP e RUP foram mencionados por apenas 1 respondente cada.

É importante notar que muitos profissionais indicaram o uso de abordagens híbridas ou adaptadas, combinando elementos de diferentes metodologias para atender às necessidades específicas de seus contextos. Exemplos de respostas incluem: “Scrum e Kanban”, “Metodologia Híbrida (Ágil com elementos do tradicional)”, “Utilizamos Scrum adaptado à nossa realidade”.

A prevalência de metodologias ágeis está de acordo com as tendências atuais da indústria de software e reforça a importância da análise de métricas específicas para esses contextos, como discutido por Leal (2023) e Ramos (2022)(LEAL, 2023; RAMOS, 2022).

4.4 Categorias de Métricas mais Adotadas

O questionário investigou o nível de utilização de quatro categorias de métricas de software e os resultados ilustrados no gráfico da Figura 8 abaixo, revelam uma abordagem de medição multifacetada por parte dos profissionais. Ao realizar uma análise detalhada da distribuição de uso para cada categoria oferece, é possível verificar os seguintes *insights*:

- **Métricas de Processo:** Essa categoria destaca-se por ter o maior número de respostas na faixa “Frequentemente utilizada”(20 respostas). Isso indica uma forte preocupação das equipes com a eficiência e a fluidez do fluxo de trabalho;
- **Métricas de Projeto e Recursos:** As métricas de Projeto (18 respostas para “Mais utilizada”) e de Recursos (18 respostas para “Mais utilizada”) demonstram ser as categorias com o mais alto nível de prioridade para os respondentes. O uso intensivo de ambas sugere uma governança robusta sobre os projetos, com um controle rigoroso sobre os pilares da gestão: escopo, custo e prazo (Projeto), bem como sobre os insumos para a sua realização, como as horas de trabalho da equipe (Recursos);
- **Métricas de Produto:** Embora essenciais para a qualidade técnica, as Métricas de Produto apresentaram uma utilização mais moderada em comparação com

as outras categorias. A maior parte das respostas se concentra em "Ocasionalmente utilizada"(14 respostas) e "Frequentemente utilizada"(13 respostas), com um número menor de equipes (7 respostas) tratando como "Mais utilizadas". Isso pode sugerir que, embora a qualidade do código seja relevante, o monitoramento de métricas como complexidade ciclomática pode não ser tão prioritário ou constante quanto o acompanhamento do processo e do projeto no dia a dia.



Figura 8 – Frequência de utilização das categorias de métricas de software

Essa distribuição sugere que, na prática das equipes pesquisadas, há um foco pragmático e dominante na gestão e execução dos projetos. A eficiência do processo, o controle sobre os recursos e o andamento do projeto parecem receber a atenção mais imediata e constante. As métricas de qualidade intrínseca do software, embora importantes, podem ser aplicadas de forma mais pontual ou em fases específicas do desenvolvimento, talvez devido à maior complexidade em sua coleta e interpretação contínua.

4.4.1 Cargo X Frequência de Utilização de Métricas de Acordo com a Categoria

Como abordado anteriormente, a diversidade de papéis em uma equipe de desenvolvimento de software implica diferentes responsabilidades e, conseqüentemente, diferentes focos de interesse no que tange à medição. Para aprofundar a compreensão

sobre como as métricas são utilizadas na prática, foi realizado um cruzamento entre alguns dos cargos com maior número de respondentes com a frequência de utilização de cada tipo de métrica. A análise revela padrões distintos e prioridades específicas de acordo com cada perfil, conforme detalhado a seguir.

4.4.1.1 Perspectiva do Desenvolvedor/Engenheiro de Software

Os profissionais que atuam diretamente na construção do código, de acordo com o exposto no gráfico da Figura 9 demonstram uma distribuição de foco bastante equilibrada, porém com ênfases claras.



Figura 9 – Frequência de utilização das categorias de métricas pelo perfil de Desenvolvedor/Engenheiro de Software

- Foco em **Processo** e **Projeto**: Para os desenvolvedores, as Métricas de Processo e de Projeto são predominantemente "Frequentemente utilizadas". Isso sugere uma forte preocupação com a eficiência do fluxo de trabalho (ex: cycle time, lead time) e com o alinhamento às restrições do projeto (prazos, escopo);
- Utilização Equilibrada de Métricas de **Produto**: As Métricas de Produto apresentam uma distribuição notavelmente uniforme entre "Mais utilizada", "Frequentemente utilizada" e "Ocasionalmente utilizada". Isso pode indicar que, embora a qualidade do código seja uma preocupação constante, a intensidade de seu monitoramento pode variar dependendo da fase do projeto ou da cultura da equipe;

- Interesse em **Recursos**: As Métricas de Recursos também mostram um uso considerável, indicando que os desenvolvedores estão atentos ao esforço e tempo despendidos em suas atividades.

4.4.1.2 Perspectiva do Gerente de Projetos

Como esperado, os Gerentes de Projetos de acordo com a análise do gráfico da Figura 10, demonstram um foco claro em métricas que tem como base o controle e o planejamento.



Figura 10 – Frequência de utilização das categorias de métricas pelo perfil de Gerente de Projetos

- Domínio das Métricas de **Projeto** e **Recursos**: Para esse perfil, as Métricas de Projeto e de Recursos são as mais utilizadas. A alta frequência na categoria "Mais utilizada" para ambas reflete a responsabilidade central desse papel em gerenciar custos, prazos e a alocação de pessoal;
- Processo como Ferramenta de Gestão: As Métricas de **Processo** são "Frequentemente utilizadas", mostrando que os gerentes vêm a otimização do fluxo de trabalho como um meio para atingir os objetivos do projeto;
- **Produto** como Resultado: As Métricas de Produto são utilizadas de forma mais moderada ("Ocasionalmente utilizada" é a categoria mais frequente), o que pode sugerir que os gerentes confiam nos especialistas (QAs e desenvolvedores) para

o monitoramento detalhado da qualidade técnica, focando mais nos resultados macro do projeto.

4.4.1.3 Perspectiva do Analista de Qualidade/QA

Os profissionais de Garantia de Qualidade (QA) de acordo com os dados expostos no gráfico da Figura 11, demonstraram um perfil de uso de métricas alinhado com sua missão de assegurar a qualidade do produto e a robustez do processo.



Figura 11 – Frequência de utilização das categorias de métricas pelo perfil de Analista de Qualidade

- Foco em **Produto** e **Processo**: A análise para os QAs é inequívoca: Métricas de Produto e Métricas de Processo são "Mais utilizadas" e "Frequentemente utilizadas" de forma predominante. Isso reflete seu duplo foco: garantir a qualidade intrínseca do software (baixa densidade de defeitos, etc.) e a eficácia do processo que previne a introdução desses defeitos (ex: cobertura de testes).
- Visão Abrangente: A utilização expressiva de Métricas de **Projeto** e **Recursos** indica que o profissional de QA moderno possui uma visão estratégica, compreendendo como as restrições de projeto e a alocação de recursos impactam a capacidade de se atingir os níveis de qualidade desejados.

4.4.1.4 Perspectiva do Scrum Master

O Scrum Master, como facilitador do processo, exibe um padrão de uso de métricas que reflete diretamente suas responsabilidades dentro do framework Scrum, o que pode ser visto no gráfico da Figura 12.



Figura 12 – Frequência de utilização das categorias de métricas pelo perfil de Scrum Master

- **Prioridade em Processo e Recursos:** As Métricas de Processo e de Recursos são as mais utilizadas por esse perfil. Isso é consistente com o papel do Scrum Master de remover impedimentos, otimizar o fluxo de trabalho da equipe (processo) e garantir que a equipe tenha os recursos necessários e um ritmo sustentável (recursos).
- **Foco Moderado em Projeto e Produto:** As Métricas de Projeto e de Produto são utilizadas de forma mais ocasional. Isso se alinha ao framework Scrum, no qual o Product Owner é o principal responsável pelo escopo e valor (ligado ao projeto) e o Time de Desenvolvimento é o responsável pela qualidade do incremento (produto), cabendo ao Scrum Master focar na saúde e eficácia do processo que permite que isso aconteça.

4.4.1.5 Análise geral

Realizando uma análise comparativa entre os diferentes perfis profissionais é possível perceber a "importância" de uma categoria de métrica de acordo com a perspectiva de cada observador. Enquanto desenvolvedores e QAs se aprofundam nas métricas de produto e processo, os gerentes de projeto priorizam o controle de custos e prazos, e os Scrum Masters focam na fluidez do trabalho da equipe.

Essa constatação reforça a necessidade de realizar a medição de uma maneira eficiente e colaborativa. Uma organização que busca a melhoria contínua não pode depender de um único conjunto de métricas, mas sim de um processo de medição estruturado e eficaz em todas as áreas. A influência das métricas no processo de desenvolvimento reside, não apenas na escolha dos indicadores corretos, mas na capacidade da organização de sintetizar essas diferentes visões em uma estratégia de qualidade unificada.

4.5 Frequência de Revisão das Métricas

Um grande indicador da maturidade de um processo de medição é a regularidade com que as métricas são analisadas e utilizadas para gerar algum tipo de *feedback*. Nesse contexto, a pesquisa investigou a frequência com que as equipes dos respondentes revisam suas métricas. Os resultados, apresentados no gráfico da Figura 13 abaixo, indicam uma diversidade de cadências de revisão, refletindo diferentes necessidades e níveis de maturidade dos processos de medição.

A análise dos resultados do gráfico acima revela alguns pontos relevantes:

- Predominância de ciclos curtos: As frequências de revisão mais citadas foram "Semanalmente"(21,7%) e "Mensalmente"(19,6%). Somadas às cadências "Diariamente"(15,2%) e "Bissemanalmente"(13%), percebe-se que a grande maioria das equipes (quase 70%) revisa suas métricas em ciclos curtos, alinhados com rituais de metodologias ágeis como o Scrum (que geralmente possui sprints semanais ou quinzenais);
- Revisão contínua vs. Revisão por evento: A existência de respostas como "a cada *commit*" ou "sempre que subimos algo para produção"(incluídas na categoria "Outros") aponta para uma prática de monitoramento contínuo, fortemente associada a ambientes de Integração e Entrega Contínua (CI/CD) e DevOps. Nesses contextos, a revisão não ocorre em uma reunião agendada, mas é um gatilho automático que valida a qualidade a cada alteração;
- Falta de padronização: O agrupamento "Outros" representa uma parcela significativa de 13% dos respondentes. Ao analisar as respostas abertas associadas

a essa categoria, foram encontrados comentários como "a frequência é muito variável", "não há periodicidade pré-definida" ou "depende da métrica", indicando que, para uma parcela das equipes, o processo de revisão ainda não está totalmente institucionalizado ou padronizado, sendo realizado de uma maneira mais reativa (somente quando é percebida a necessidade de alguma alteração).

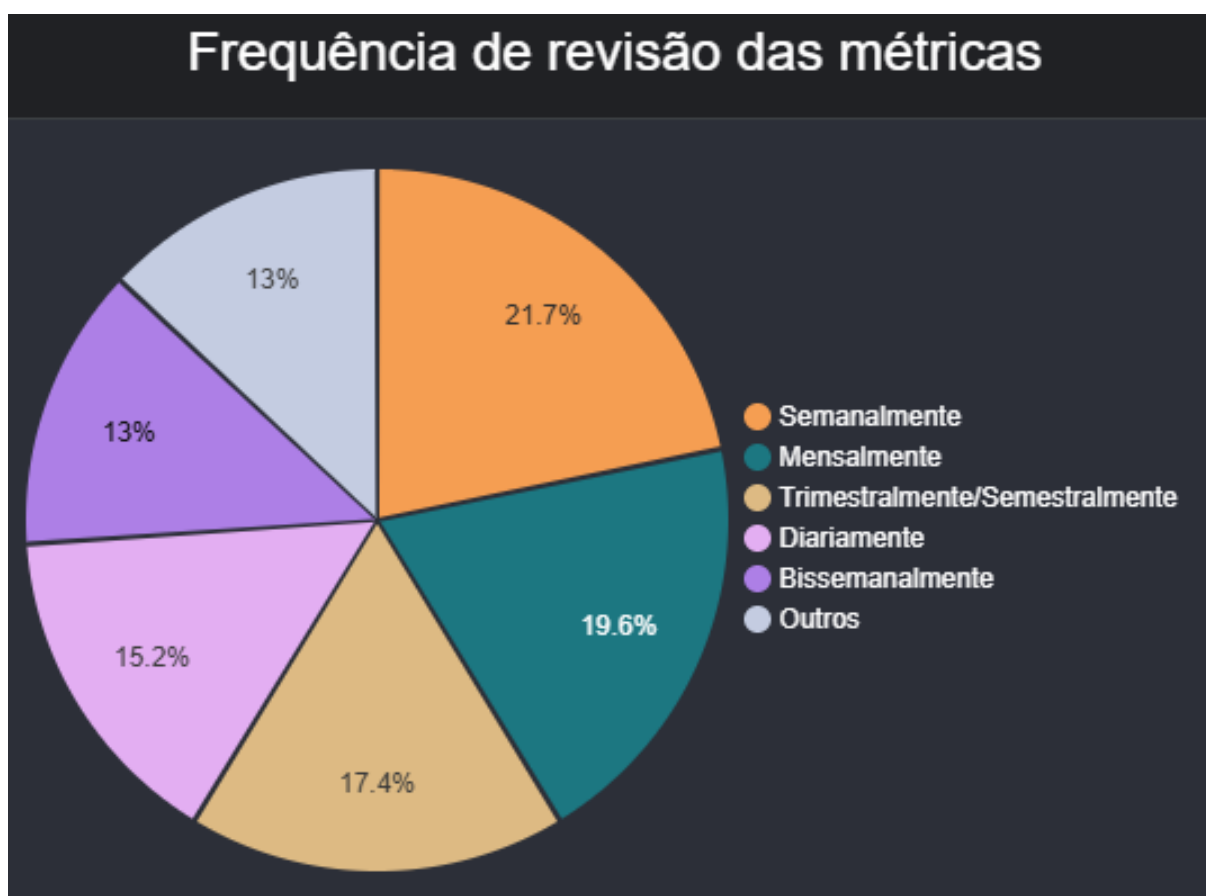


Figura 13 – Frequência de revisão das métricas de qualidade

Esses dados reforçam a forte influência das metodologias ágeis no dia a dia das equipes. A tendência de revisões semanais e mensais está em sintonia com a necessidade de inspeção e adaptação contínuas, pregadas por metodologias de desenvolvimento como o Scrum. No entanto, a presença de respostas que indicam falta de padronização revela uma oportunidade de melhoria.

4.6 Contribuição das Métricas na Tomada de Decisões

Uma das premissas fundamentais para a adoção de métricas de software é a capacidade de evoluir de uma gestão reativa para um modelo de tomada de decisão fundamentado em evidências quantitativas. Com o intuito de verificar a aplicação prática

dessa premissa, foi investigado, por meio do questionário, a percepção dos profissionais sobre a contribuição efetiva das métricas no direcionamento estratégico de suas organizações. Os resultados obtidos revelam uma tendência altamente positiva.

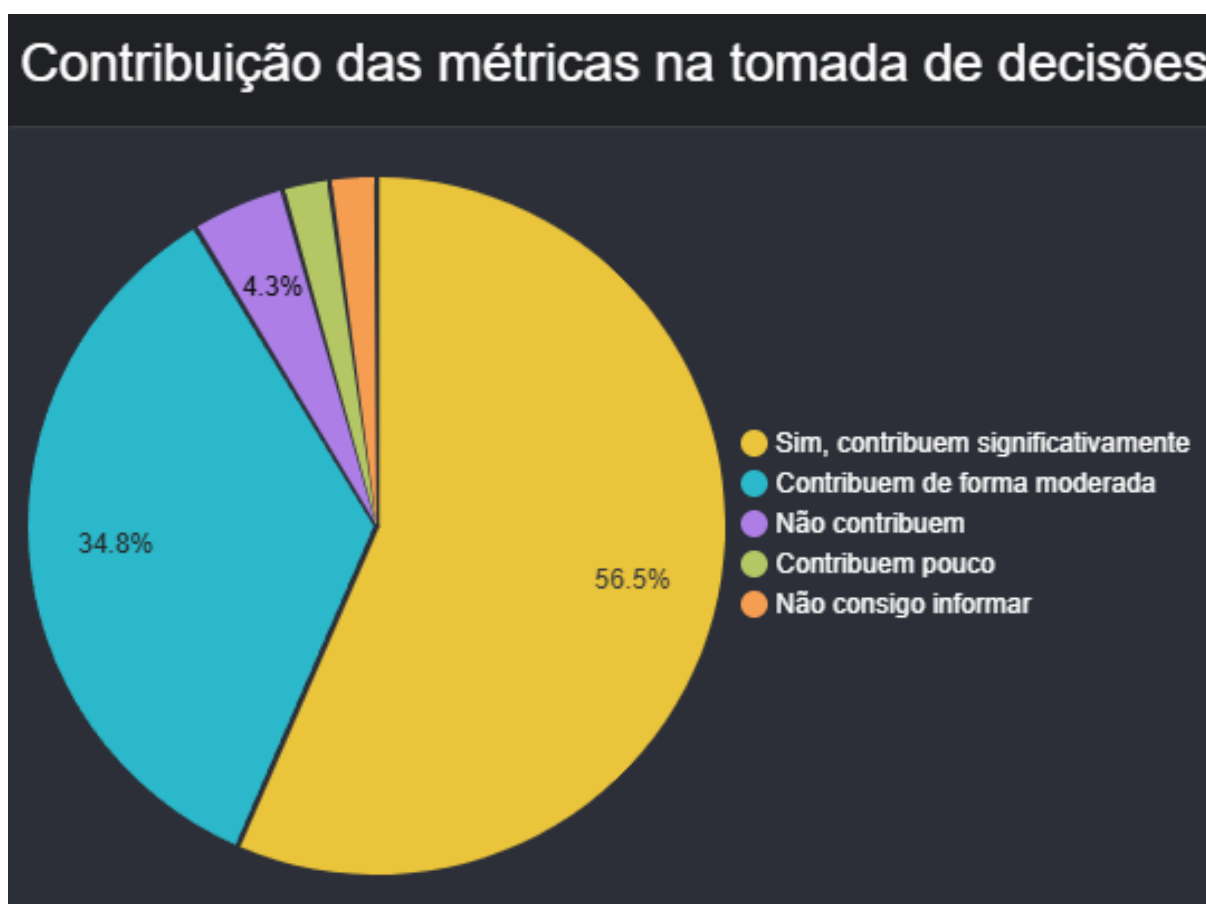


Figura 14 – Percepção dos profissionais sobre a contribuição das métricas na tomada de decisões estratégicas

A análise do gráfico da Figura 14 acima revela que a grande maioria dos respondentes enxerga uma conexão clara entre medição e a tomada de decisões estratégicas:

- Grande impacto positivo: Mais de 90% dos participantes afirmam que as métricas contribuem, em algum grau, para as decisões estratégicas. Esse resultado sugere que no caso dessas equipes as métricas provavelmente deixaram de ser apenas ferramentas operacionais de controle de qualidade e passaram a ser vistas como instrumentos valiosos para o planejamento dos projetos;
- Relevância estratégica: A percepção de que as métricas contribuem "significativamente", indica que, para mais da metade das equipes representadas, os dados de qualidade não são apenas informativos, mas sim influentes. Elas podem estar sendo utilizadas para justificar investimentos em tecnologia, direcionar o desenvolvimento de novas linhas de produto, alocar recursos em áreas mais críticas ou até mesmo redefinir metas de negócio;

- **Minoria com baixo impacto:** Apenas uma pequena fração dos respondentes (4,3%) percebe uma contribuição baixa ou nula. Esse grupo minoritário pode estar inserido em contextos organizacionais em que a cultura de dados ainda é incipiente, ou em que as métricas coletadas não estão devidamente alinhadas aos objetivos estratégicos, tornando-se mais uma formalidade do que uma ferramenta de gestão.

Portanto, os dados expostos acima reforçam uma das teses centrais deste trabalho: as métricas, quando bem aplicadas, possuem um papel transformador que vai além da detecção de defeitos, se tornando um elo fundamental entre a operação técnica e a estratégia de negócio, permitindo que as organizações tomem decisões cada vez mais inteligentes e baseadas em dados reais.

4.7 Análise dos Resultados

A análise dos dados coletados de 46 profissionais da área de software permitiu extrair uma visão pragmática sobre a utilização de métricas na indústria. Alguns dos pontos mais relevantes descobertos foram:

4.7.1 Perfil dos Profissionais e Contexto Metodológico

- **Diversidade de Papéis:** A pesquisa contou com uma amostra heterogênea, com destaque para Desenvolvedores/Engenheiros de Software (26,1%), Gerentes de Projetos (19,6%) e Analistas de Qualidade/QA (15,2%). Essa diversidade permitiu uma análise multifacetada do tema;
- **Domínio Ágil:** Houve uma forte predominância de metodologias ágeis, com o Scrum sendo o framework mais citado, frequentemente em combinação com o Kanban. Isso estabelece o contexto de que as práticas de medição atuais estão inseridas em ambientes de desenvolvimento iterativo e de feedback rápido.

4.7.2 Práticas de Medição: Ferramentas e Frequência

- **Alta Adoção de Ferramentas:** A grande maioria das equipes (cerca de 87%) utiliza ferramentas para a coleta de métricas. A combinação de ferramentas de gestão de projetos e de análise estática é a abordagem mais comum (37,8%), indicando uma busca por uma visão integrada dos dados;
- **Frequência Alinhada ao Ágil:** A revisão de métricas ocorre, na maioria das vezes, em ciclos curtos. As frequências "Semanalmente" (21,7%) e "Mensalmente" (19,6%)

são as mais comuns, o que é consistente com a cadência de rituais ágeis como *sprints* e reuniões de planejamento.

4.7.3 Utilização e Priorização das Categorias de Métricas

- Foco em Gestão e Execução: A análise da frequência de uso revelou um foco dominante em **Métricas de Projeto** e **Métricas de Recursos**, que foram apontadas como as "Mais utilizadas" pela maioria dos respondentes. Isso indica uma forte prioridade no controle de prazos, custos e alocação de equipe;
- Importância do Processo: As **Métricas de Processo** são "Frequentemente utilizadas" pela maior parte dos profissionais, refletindo a preocupação com a eficiência do fluxo de trabalho e a remoção de gargalos, em linha com as práticas ágeis;
- Uso Moderado de Métricas de Produto: As **Métricas de Produto**, embora cruciais para a qualidade técnica, apresentaram um uso mais moderado, com a maioria das respostas concentradas em "Ocasionalmente utilizada".

4.7.4 Perspectivas Diferentes por Cargo Profissional

- Gerentes de Projeto: Possuem um maior foco em **Métricas de Projeto** e **Recursos**, alinhado com suas responsabilidades de gestão;
- Analistas de Qualidade (QA): Demonstram um foco claro e equilibrado em **Métricas de Produto** e **Processo**, refletindo sua missão de garantir a qualidade do software e a eficácia dos testes;
- Desenvolvedores e Scrum Masters: Mostraram uma maior priorização de **Métricas de Processo**, buscando otimizar o fluxo de trabalho e a eficiência da equipe.

4.7.5 Impacto Estratégico das Métricas

- Contribuição Significativa para Decisões: Um dos achados mais relevantes da pesquisa foi que a esmagadora maioria dos profissionais (91,3%) percebe que as métricas contribuem de forma moderada ou significativa para a tomada de decisões estratégicas na organização;
- Métricas como Ativo Estratégico: Isso confirma a hipótese de que as métricas transcenderam um nível puramente operacional e evoluíram para se tornarem insumos valiosos que conectam a área técnica aos objetivos de negócio, influenciando investimentos, planejamento e alocação de recursos.

Em síntese, a pesquisa de campo realizada revelou que, embora exista um forte foco prático na gestão de projetos e recursos, as métricas são amplamente reconhecidas como ferramentas estratégicas e essenciais para a melhoria contínua, com sua aplicação e priorização variando de acordo com a perspectiva de cada papel profissional dentro da equipe.

5 CONSIDERAÇÕES FINAIS

Este capítulo final consolida os resultados e as reflexões desenvolvidas ao longo deste trabalho, sintetizando os principais achados, respondendo formalmente os objetivos propostos e discutindo as implicações práticas e teóricas do estudo. Adicionalmente, são expostas as limitações da pesquisa e sugeridos caminhos para trabalhos futuros, visando contribuir para a contínua evolução do conhecimento na área de métricas e qualidade de software.

5.1 Retomada da Pesquisa e Síntese dos Resultados

Este trabalho foi motivado pela necessidade de compreender como as métricas de software, embora amplamente reconhecidas como ferramentas valiosas, podem efetivamente influenciar a melhoria contínua da qualidade de software e dos processos de desenvolvimento na prática dentro da indústria de software (SOUZA, 2016). Embora a literatura apresente um vasto leque de métricas e modelos de qualidade, persiste uma lacuna entre a coleta de dados e sua aplicação estratégica para gerar algum tipo de aprimoramento real (LEAL, 2023; SOUZA, 2016). A presente pesquisa buscou investigar essa dinâmica, combinando uma revisão teórica com uma pesquisa de campo junto a profissionais da área para obter uma visão pragmática do cenário atual.

A análise dos dados e da literatura permitiu consolidar os seguintes achados:

- **Consolidação das Práticas Ágeis:** A pesquisa de campo confirmou a predominância de metodologias ágeis (Scrum, Kanban) no cenário atual. Esse contexto reforça a importância de métricas alinhadas a ciclos de *feedback* rápidos, que permitem a inspeção e adaptação contínuas, conforme discutido no referencial teórico (RAMOS, 2022; KANTORSKI; KROTH, 2018);
- **Reconhecimento do Valor Estratégico das Métricas:** Um dos resultados mais expressivos da pesquisa foi a constatação de que a esmagadora maioria dos profissionais (mais de 91%) enxerga as métricas não apenas como ferramentas operacionais para controle de defeitos, mas como ativos que contribuem de forma moderada ou significativa para a tomada de decisão estratégica. Isso corrobora a visão de que a qualidade de software é um fator competitivo fundamental (Avila 2024, Humphrey 1996);
- **Visão Multifacetada da Medição:** Ficou evidente que a utilização de métricas varia significativamente conforme o papel do profissional. Gerentes de Projeto

focam em custo e prazo (**Métricas de Projeto e Recursos**), enquanto Analistas de QA e Desenvolvedores se aprofundam em indicadores técnicos (**Métricas de Produto e Processo**). Essa diversidade de perspectivas, embora natural, reforça a necessidade de uma abordagem holística na seleção e análise de métricas;

- O Fator Humano e a Clareza de Objetivos: A pesquisa de campo revelou uma forte correlação positiva entre a clareza dos objetivos de um programa de medição e a satisfação da equipe com seu uso. Equipes que compreendem claramente o propósito das métricas e como interpretá-las tendem a perceber maior valor em sua aplicação, sugerindo que os aspectos culturais e de comunicação são tão importantes quanto os técnicos na implementação bem-sucedida de um programa de métricas;
- Integração de Ferramentas e Percepção de Valor: A análise correlacional revelou que equipes que utilizam tanto ferramentas de análise estática quanto de gestão de projetos apresentam a maior taxa de percepção de "contribuição significativa" das métricas (70,6%), indicando que a combinação de diferentes perspectivas de medição potencializa o valor percebido.

5.2 Resposta aos Objetivos da Pesquisa

O percurso metodológico e a análise dos resultados permitiram alcançar os objetivos delineados para este trabalho.

O objetivo geral deste trabalho foi analisar a influência das métricas de software na qualidade de desenvolvimento de projetos, investigando como sua aplicação sistemática contribui para a eficiência nos processos e para o aprimoramento do produto final. Este objetivo foi alcançado por meio da combinação de uma revisão bibliográfica abrangente e uma pesquisa de campo com 46 profissionais da área, que permitiu identificar padrões, correlações e *insights* sobre como as métricas efetivamente impactam os processos de desenvolvimento em contextos reais.

Os Objetivos Específicos foram igualmente alcançados:

- Identificação e análise das principais métricas: Por meio da revisão da literatura e da pesquisa de campo, foram identificadas e analisadas as principais métricas relevantes para avaliação e aprimoramento dos processos, categorizadas em métricas de produto, processo, projeto e recursos;
- Realização da pesquisa de campo: Foi conduzida com sucesso uma pesquisa por meio de formulário online distribuído para profissionais da área, obtendo 46

respostas válidas que forneceram *insights* valiosos sobre práticas reais de medição em diferentes contextos organizacionais e metodológicos;

- Seleção de métricas relevantes: Com base na literatura e nas respostas do formulário, foram identificadas as métricas consideradas mais relevantes para cada contexto e papel profissional. A análise correlacional permitiu compreender como diferentes perfis priorizam diferentes categorias de métricas e como essa priorização se relaciona com a percepção de valor e com o impacto na qualidade;
- Investigação da utilização de *insights* para melhoria: Foi possível mapear como os *insights* derivados da análise de métricas são utilizados para direcionar ações de melhoria, identificando que equipes com maior maturidade na utilização de métricas conseguem estabelecer um ciclo no qual a medição efetivamente orienta intervenções no processo e validar sua eficácia.

5.3 Implicações do Estudo

As descobertas desta monografia geram implicações significativas tanto para o ambiente profissional quanto para o acadêmico.

- Implicações Práticas (para a Indústria) Para gestores, desenvolvedores e equipes de QA, este estudo pode oferecer as seguintes recomendações:
 - Institucionalizar a Definição de Objetivos: Antes de implementar qualquer ferramenta ou coletar qualquer dado, as equipes devem investir tempo na definição de objetivos claros, utilizando abordagens como o GQM (Goal-Question-Metric). É fundamental que todos os membros da equipe compreendam por que uma métrica está sendo coletada e qual decisão ela irá suportar;
 - Promover uma Cultura de Métricas Construtiva: As métricas devem ser posicionadas como ferramentas para o aprendizado e a melhoria do processo e não para a avaliação ou punição individual;
 - Criar uma Visão Holística e Integrada: Dada a constatação de que cada papel profissional prioriza um conjunto diferente de métricas, surge a necessidade de criar *dashboards* e relatórios que integrem as diferentes categorias (**Métricas de Produto, Processo, Projeto ou Recursos**). Uma visão unificada permite que, por exemplo, a equipe de desenvolvimento entenda o impacto de seu trabalho técnico nas metas do projeto e que os gestores compreendam como as restrições de prazo e custo afetam a qualidade do produto final.

- Contribuições Acadêmicas (para a Pesquisa) Para a academia, este trabalho contribui em diversos aspectos, sendo eles:
 - Modelo Correlacional: A análise correlacional entre perfil profissional, metodologia adotada e percepção de valor das métricas oferece um modelo conceitual que pode ser expandido e testado em pesquisas futuras.
 - Integração de Perspectivas: O trabalho integra perspectivas da Engenharia de Software, gestão de projetos e garantia de qualidade, oferecendo uma visão multidisciplinar que enriquece a compreensão do tema.
 - Validação da Importância dos Fatores Humanos: Ao demonstrar a forte correlação entre clareza de objetivos e satisfação, o estudo reforça a necessidade de que futuras pesquisas sobre métricas incorporem análises sobre fatores culturais, de comunicação e de gestão;
 - Análise por Papéis Profissionais: A segmentação da análise por cargo oferece uma taxonomia útil para entender as diferentes prioridades e "dores" de cada profissional no que tange à qualidade, abrindo um campo para estudos mais aprofundados sobre a colaboração e os conflitos de interesse na gestão da qualidade.

5.4 Limitações da Pesquisa

Apesar dos resultados obtidos, é importante reconhecer as limitações inerentes a este estudo, o que demonstra honestidade intelectual e abre caminhos para futuras investigações.

- Tamanho e escopo da amostra: A pesquisa contou com 46 respondentes. Embora seja uma amostra diversificada, ela não permite a generalização estatística dos resultados para toda a indústria de software brasileira. Os achados representam um retrato do grupo pesquisado, que pode ter sido influenciado pelo alcance das redes em que o questionário foi divulgado;
- Natureza dos dados: Os dados foram coletados por meio de um questionário de autorrelato. Essa abordagem está sujeita a vieses de percepção, em que as respostas podem refletir a visão idealizada dos participantes sobre suas práticas e não necessariamente a realidade objetiva;
- Profundidade da análise: O uso de um questionário como principal instrumento permite uma visão ampla e quantitativa do cenário. No entanto, ele não possibilita o aprofundamento em contextos organizacionais específicos, em motivações individuais ou nas nuances dos processos, algo que um estudo de caso qualitativo com entrevistas poderia proporcionar.

5.5 Sugestões para Trabalhos Futuros

A partir das conclusões e limitações deste trabalho, emergem diversas oportunidades para futuras pesquisas que possam aprofundar e expandir o conhecimento sobre o tema:

- Estudo de caso qualitativo: Sugere-se a realização de um estudo de caso mais aprofundado em uma ou mais empresas para investigar qualitativamente como os *insights* gerados pelas métricas são discutidos, priorizados e efetivamente convertidos em ações de melhoria de processo;
- Pesquisa quantitativa em larga escala: Uma futura pesquisa poderia aplicar o questionário a uma amostra maior e estatisticamente significativa, segmentada por setores da indústria (ex: *fintechs*, *e-commerce*, saúde), para identificar se existem padrões de uso de métricas específicos para cada domínio;
- Validação Quantitativa de Impacto: Pesquisas que estabeleçam correlações estatisticamente significativas entre a adoção de métricas específicas e indicadores objetivos de qualidade e eficiência, como redução de defeitos, tempo de ciclo ou satisfação do cliente;
- Desenvolvimento de *Frameworks* Adaptáveis: Criação e validação de *frameworks* que auxiliem organizações a selecionar e implementar métricas adequadas ao seu contexto específico, considerando fatores como metodologia, tamanho da equipe, domínio de aplicação e objetivos estratégicos;
- Exploração do Fator Humano: Investigações mais aprofundadas sobre como aspectos culturais, psicológicos e organizacionais influenciam a eficácia dos programas de métricas, incluindo resistências, motivações e fatores que promovem a adoção;
- Integração com **Inteligência Artificial (IA)**: Estudos sobre como técnicas de IA e aprendizado de máquina podem potencializar a análise de métricas, identificando padrões não evidentes e fornecendo recomendações proativas para melhoria de processos.

5.6 Conclusão

Resumindo, este trabalho partiu da observação de uma diferença entre o potencial teórico das métricas de software e a sua aplicação prática. Por meio de uma pesquisa realizada com profissionais da área de software, foi concluído que essa ponte vem sendo construída de forma sólida na indústria. Os resultados da pesquisa também

evidenciaram que as métricas vão além de seu papel técnico, configurando-se como componentes estratégicos, que quando alinhados a objetivos claros e a uma cultura de colaboração, podem influenciar positivamente a melhoria contínua dos processos de desenvolvimento, reforçando que a busca pela qualidade de software é um processo em constante atualização, que envolve várias frentes: a medição objetiva, a inteligência humana e uma comunicação eficaz, que juntas se tornam os principais fatores para entregar produtos com mais valor.

Referências

- ALVES, J. M. et al. Qualidade de Software do Maior Sistema de Teletoxicologia Utilizado no Brasil. In: *Anais do XXXVI Congresso da Sociedade Brasileira de Computação*. Porto Alegre: Sociedade Brasileira de Computação (SBC), 2016. p. 1475–1484. 27, 28
- BONA, R. S. D. Mensuração de capacidade operacional e alocação de recursos humanos nas macrofunções da controladoria pública municipal brasileira. *Sociedade, Contabilidade e Gestão*, 2022. Acesso em: 14 jun. 2025. Disponível em: <<https://revistas.ufrj.br/index.php/scg/article/view/41655>>. 30
- CMMI Institute. *CMMI Performance Summary Report 2019-2021*. [S.l.], 2025. Acesso em: 20 de Maio 2025. Disponível em: <<https://cmmiinstitute.com/getattachment/738104c0-a6f0-4e1c-8bbe-35076b75f36e/attachment.aspx>>. 32
- CONCEIÇÃO, I. T. d. PRÁTICAS DE SEGURANÇA NO DESENVOLVIMENTO COM O USO DE METODOLOGIAS ÁGEIS. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, 2024. Acesso em: 20 de Maio 2025. Informações sobre volume, número e páginas não disponíveis na referência fornecida. Disponível em: <https://www.semanticscholar.org/paper/PR%C3%81TICAS-DE-SEGURAN%C3%87A-NO-DESENVOLVIMENTO-COM-O-USO-Concei%C3%A7%C3%A3o/4f60319f503c0e644c2f40bc9e4fb782716675ad?utm_source=direct_link>. 23
- FENTON, N.; BIEMAN, J. *Software Metrics: A Rigorous and Practical Approach*. 3. ed. CRC Press, 2014. Acesso em: 5 de Maio 2025. Disponível em: <https://api.pageplace.de/preview/DT0400.9781439838235_A38252996/preview-9781439838235_A38252996.pdf>. 21, 22
- FISCHER, G. S. et al. Combinando elasticidade proativa e reativa para gestão multi-hospitalar de recursos humanos no contexto das cidades inteligentes. In: *Anais do XXIV Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS)*. Porto Alegre, RS, Brasil: Sociedade Brasileira de Computação, 2024. p. 249–260. ISSN 2763-8952. Acesso em: 14 jun. 2025. Disponível em: <<https://doi.org/10.5753/sbcas.2024.2179>>. 30
- GAMBACORTA, L. et al. *Generative AI and labour productivity: a field experiment on coding*. [S.l.], 2024. Acesso em: 18 de Abril 2025. Disponível em: <<https://www.bis.org/publ/work1208.html>>. 25
- GIL, A. C. *Métodos e técnicas de pesquisa social*. 6. ed. São Paulo: Atlas, 2008. 35
- HERNANDEZ, V. H.; GOMES, F. i. n. c. et al. O papel do Product Owner no framework Scrum: relação entre metodologias ágeis, qualidade no desenvolvimento de softwares e gestores de projetos. *Brazilian Journal of Development*, 2023. Acesso em: 20 de Maio 2025. Informações sobre volume, número e páginas não disponíveis na referência fornecida. Disponível em: <https://www.semanticscholar.org/paper/O-papel-do-Product-Owner-no-framework-Scrum%3A-entre-Hernandez-Gomes/ff2e0b1bee96a7144f771d4a3788f444c5ccc4e6?utm_source=direct_link>. 23

- HUMPHREY, W. S. *Introduction to the Personal Software Process*. [S.l.]: Addison-Wesley, 1996. 21, 32
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC 25040:2011 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process*. Genebra, 2011. 21
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC 25022:2016 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of quality in use*. Genebra, 2016. 22
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC 25010:2023 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model*. Genebra, 2023. 20, 26, 27
- ISO/IEC 25000 Users Group. *ISO/IEC 25010 - System and software quality models*. 2024. <<https://www.iso25000.com/index.php/en/iso-25000-standards/iso-25010>>. 12, 21
- KANER, C.; FALK, J. H.; NGUYEN, H. Q. *Testing Computer Software*. Second. New York: John Wiley & Sons, 1999. 22
- KANTORSKI, G.; KROTH, M. Controle de métricas no processo de desenvolvimento de software através de uma ferramenta de workflow. In: *Anais do Salão de Iniciação Científica da SETREM*. [s.n.], 2018. Acesso em: 20 de Abril 2025. Disponível em: <<https://www.ufsm.br/app/uploads/sites/350/2018/07/2540.pdf>>. 26, 27, 28, 55
- LEAL, S. d. S. *Desenvolvimento de um modelo de seleção e avaliação de métricas de software para contextos ágeis*. Dissertação (Mestrado) — Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2023. 16, 17, 43, 55
- LENCIONI, P. *The Five Dysfunctions of a Team: A Leadership Fable*. San Francisco: Jossey-Bass, 2012. 33
- LIMA, V. R. d. *DETERMINANTES DOS ÍNDICES DE RENTABILIDADE: ROA, ROE, ROI*. Monografia (Trabalho de Conclusão de Curso (Graduação em Ciências Contábeis)), Uberlândia, 2017. Acesso em: 6 de Junho de 2025. Disponível em: <<https://repositorio.ufu.br/bitstream/123456789/19969/1/Determinantes%C3%8DndicesRentabilidade.pdf>>. 29
- MCCABE, T. J. A Complexity Measure. *IEEE Transactions on Software Engineering*, SE-2, n. 4, p. 308–320, 1976. Acesso em: 15 de Abril 2025. Disponível em: <<http://dx.doi.org/10.1109/TSE.1976.233837>>. 24
- MEDEIROS, G. G. *Métodos e Práticas de Estimativas de Prazos E Custos em Projetos Ágeis*. Monografia (Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação)), Uberlândia, 2023. Acesso em: 6 de Junho de 2025. Disponível em: <<https://repositorio.ufu.br/bitstream/123456789/38733/1/M%C3%A9todosPr%C3%A1ticasEstimativas.pdf>>. 27, 28

- Microsoft Corporation. *Code metrics - Cyclomatic complexity*. 2024. <<https://learn.microsoft.com/en-us/visualstudio/code-quality/code-metrics-cyclomatic-complexity>>. Acesso em: 18 de Abril 2025. 24
- MILLS, E. E. *Software Metrics*. Pittsburgh, 1988. (SEI Curriculum Module SEI-CM-12-1.1). 16
- MRIDUL, B.; RANA, A. Key software metrics and its impact on each other for software development projects. *SIGSOFT Software Engineering Notes*, v. 41, n. 1, p. 1–4, jan. 2016. Acesso em: 13 jun. 2025. Disponível em: <<https://doi.org/10.1145/2853073.2853087>>. 29
- NASA Engineering and Safety Center. *Cyclomatic complexity and basis path testing study*. Hampton, 2020. Acesso em: 27 de Abril 2025. Disponível em: <<https://ntrs.nasa.gov/api/citations/20205011566/downloads/20205011566.pdf>>. 25
- NEVES, L. F. S. *INDICADORES ECONÔMICO-FINANCEIROS E A PANDEMIA DE COVID-19: análise dos setores mais afetados pela pandemia*. Monografia (Trabalho de Conclusão de Curso (Graduação em Ciências Contábeis)), Uberlândia, 2022. Acesso em: 6 de Junho de 2025. Disponível em: <<https://repositorio.ufu.br/bitstream/123456789/35609/1/IndicadoresEcon%C3%B4micoFinanceiros.pdf>>. 29
- PINHEIRO, F. V. et al. Utilizando teoria das filas e simulação de alocação de recursos em ambientes de desenvolvimento de software. In: *Anais do V Workshop em Modelagem e Simulação de Sistemas Intensivos em Software (MSSIS)*. Porto Alegre, RS, Brasil: Sociedade Brasileira de Computação, 2023. p. 51–60. Acesso em: 13 jun. 2025. Disponível em: <<https://doi.org/10.5753/mssis.2023.235580>>. 29, 31
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2016. 33
- RAHMAN, D. N. I.; ANDRESWARI, R.; FAUZI, R. Implementation of robotic process automation for sentiment analysis of lecturer evaluation by students (edom) at higher education information system. In: *2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*. [s.n.], 2022. p. 01–06. Acesso em: 13 jun. 2025. Disponível em: <<https://ieeexplore.ieee.org/document/10037497>>. 30
- RAMOS, R. A. d. L. *Monitoramento de métricas de qualidade e produtividade em projetos ágeis de software através da integração de dados extraídos de ferramentas de gestão e testes*. Dissertação (Mestrado) — Programa de Pós-Graduação em Tecnologia da Informação, Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, João Pessoa, 2022. 16, 42, 43, 55
- ROSENBERG, L.; HAMMER, T.; SHAW, J. SOFTWARE METRICS AND RELIABILITY. In: *Proceedings of the 9th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE Computer Society, 1998. Acesso em: 20 de Abril 2025. Disponível em: <<https://citeseerx.ist.psu.edu/pdf/31e3f5732a7af3aec364b6cc2a85d9495b5c159>>. 26
- SERPRO. *Metodologia de Gestão de Riscos e Controles Internos*. Brasília, 2023. Acesso em: 3 de Maio de 2025. Disponível em: <<https://www.transparencia.gov.br>>.

- serpro.gov.br/governanca/governanca-corporativa/gestao-de-riscos/dd-ri-001-2023_merged.pdf>. 22
- SHAH, H. M. et al. Mining and fusing productivity metrics with code quality information at scale. In: *2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. Bogotá, Colombia: [s.n.], 2023. p. 563–567. Acesso em: 12 jun. 2025. Disponível em: <<https://ieeexplore.ieee.org/document/10336313>>. 29, 30
- SILVA, R. A. et al. UMA ABORDAGEM BASEADA EM MÉTRICAS PARA AVALIAÇÃO DA QUALIDADE DE PROJETOS DE SOFTWARE. *REVISTA CIENTÍFICA DA UNIFENAS*, 2024. Acesso em: 19 de Abril 2025. Disponível em: <<https://www.semanticscholar.org/paper/UMA-ABORDAGEM-BASEADA-EM-M%C3%89TRICAS-PARA-AVALIA%C3%87%C3%83O-DA-Silva-Filho/6c517ab87d6ba210e3149aa52433acddd8ade9b3>>. 25
- SOARES, E.; ZAIDAN, F.; JAMIL, G. Demonstrativo financeiro do projeto de software associado às métricas do plano de carreira profissional atrelado à qualidade do processo de desenvolvimento de software nas empresas de tecnologia da informação. In: *Proceedings of the 12th CONTECSI - International Conference on Information Systems and Technology Management*. TECSI, 2015. ISSN 2448-1041. Acesso em: 12 jun. 2025. Disponível em: <<https://www.contecsi.tecsi.org/index.php/contecsi/12CONTECSI/paper/view/1909>>. 29
- SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 33
- SonarSource. *SonarQube*. 2006. <<https://www.sonarsource.com/products/sonarqube/>>. Acesso em: 20 de Maio 2025. 25
- SOUZA, P. P. d. *A Utilidade dos Valores Referência de Métricas na Avaliação da Qualidade de Softwares Orientados por Objeto*. Dissertação (Mestrado) — Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2016. 16, 17, 55
- SYNOVIC, N. M. et al. Snapshot metrics are not enough: Analyzing software repositories with longitudinal metrics. In: *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE '22)*. Association for Computing Machinery, 2022. Acesso em: 15 jun. 2025. Disponível em: <<https://dl.acm.org/doi/10.1145/3551349.3559517>>. 30, 31
- TRODO, L. D. *Uso de Métricas nos Testes de Software*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, 2009. Acesso em: 18 de Abril 2025. Disponível em: <<http://hdl.handle.net/10183/18574>>. 25
- VALLEJO, P. et al. Métricas de productividad y gestión del trabajo basadas en psp para estudiantes de ingeniería de sistemas: caso universidad eafit. *ReCIBE, Revista electrónica de Computación, Informática, Biomédica y Electrónica*, v. 12, n. 2, p. C10–C12, 2024. Acesso em: 17 jun. 2025. Disponível em: <<https://recibe.cucei.udg.mx/index.php/ReCIBE/article/view/333>>. 31
- ÁVILA, A. et al. Impacto das Métricas de Software no Índice de Manutenção: Uma Revisão Sistemática. *Revista Científica da UNIFENAS*, v. 6, n. 5, p. 1–11, 2024. 16, 42

A Resultados da Pesquisa

Neste apêndice, disponibiliza-se o acesso à planilha contendo todos os dados brutos coletados por meio do questionário aplicado aos 46 profissionais, que serviram de base para as análises apresentadas no Capítulo 4.

O acesso pode ser feito pelo seguinte link:

Link para a Planilha de Dados no Google Sheets¹

¹ <https://docs.google.com/spreadsheets>

B Instrumento de Coleta de Dados

Este apêndice contém a versão integral do questionário utilizado como instrumento para a coleta de dados desta pesquisa, conforme descrito no capítulo 3 - Metodologia.

Pesquisa sobre o uso de métricas e sua influência na qualidade de projetos de software

Este formulário faz parte de uma pesquisa acadêmica para a elaboração do Trabalho de Conclusão de Curso (TCC), cujo objetivo é investigar a Análise de Métricas de Software. A proposta é compreender como as métricas são utilizadas para avaliar a qualidade dos processos de desenvolvimento de software nas equipes. A pesquisa busca identificar quais métricas são aplicadas, de que forma elas contribuem para a melhoria da qualidade do software e como influenciam as decisões estratégicas e operacionais da equipe.

Observação 1: Suas respostas são confidenciais e serão utilizadas apenas de forma agregada e anônima para fins desta pesquisa acadêmica.

Observação 2: Caso queira receber o resultado dessa pesquisa, envie um email para: tiagog.pinho@ucsal.edu.br

1. Qual o seu nome?

2. Qual o seu cargo atual?

Marcar apenas uma oval.

Desenvolvedor/Engenheiro de Software

Analista de Qualidade/QA

Gerente de Projetos

Arquiteto de Software

Scrum Master

Product Owner

Outro: _____

3. Sua equipe utiliza métricas de qualidade de forma sistemática em seus projetos?

Marcar apenas uma oval.

Sim

Não

Outro: _____

4. Você concorda que a utilização de métricas de qualidade é essencial para melhorar os processos de desenvolvimento de software?

Marcar apenas uma oval.

Concordo totalmente

Concordo

Discordo

Discordo totalmente

Não consigo informar

5. Com que frequência as métricas de qualidade são revisadas e analisadas em seu ambiente de trabalho

Marcar apenas uma oval.

Diariamente

Semanalmente

Bissemanalmente

Mensalmente

Trimestralmente/Semestralmente

Anualmente

Outro: _____

6. Classifique as categorias de métricas abaixo, de acordo com a frequência de utilização das mesmas pela sua equipe:

Marcar apenas uma oval por linha.

	Mais utilizada	Frequentemente utilizada	Ocasionalmente utilizada	Não utilizada
Métricas de Produto (ex.: Complexidade Ciclométrica, LOC, Densidade de Defeitos)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Métricas de Processo (ex.: Tempo de Ciclo, Tempo de Espera)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Métricas de Projeto (ex.: Custo, Prazo, Esforço)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Métricas de Recursos (ex.: Horas de trabalho da equipe)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. Como sua equipe coleta e utiliza o feedback dos clientes para medir a qualidade do software entregue? (Marque todas as opções que se aplicam)

Marque todas que se aplicam.

- Não coletamos feedback formalmente
- Coletamos feedback informalmente (e-mails, reuniões, etc.)
- Utilizamos ferramentas de pesquisa (NPS, CSAT, etc.)
- O feedback é analisado e usado para orientar melhorias no produto
- Temos um processo estruturado e contínuo de feedback → ação
- Outro: _____

8. Qual o nível de influência que as métricas relacionadas ao cliente (ex: NPS, taxa de churn, satisfação pós-entrega) possuem na definição de prioridades da equipe?

Marcar apenas uma oval.

- Nenhuma influência
- Influência ocasional, dependendo do projeto
- Usamos essas métricas como parte do processo de priorização
- São métricas principais na tomada de decisão

9. Sua equipe utiliza ferramentas de análise estática e/ou ferramentas de gestão de projetos para obter/coletar métricas de qualidade?

Marcar apenas uma oval.

- Sim, ferramentas de análise estática
- Sim, ferramentas de gestão de projetos
- Sim, ambas
- Não utilizamos ferramentas específicas
- Outro: _____

10. Na sua opinião, as métricas de qualidade contribuem efetivamente para a tomada de decisões estratégicas em sua organização?

Marcar apenas uma oval.

- Sim, contribuem significativamente
- Contribuem de forma moderada
- Contribuem pouco
- Não contribuem
- Não consigo informar

11. Quais são as principais razões para utilizar métricas de qualidade na sua equipe? (Selecione todas as que se aplicam)

Marque todas que se aplicam.

- Identificar e reduzir defeitos
- Melhorar processos internos
- Monitorar desempenho da equipe
- Cumprir requisitos de certificações/padrões
- Justificar decisões estratégicas
- Outro: _____

12. Como você avalia a satisfação da sua equipe com as métricas de qualidade atualmente utilizadas?

Marcar apenas uma oval.

- Muito satisfeito(a)
- Satisfeito
- Insatisfeito
- Muito insatisfeito(a)
- Não consigo informar

13. Na sua opinião, qual categoria de métricas tem maior impacto na melhoria da qualidade do software?

Marcar apenas uma oval.

- Métricas de Produto (ex.: Defeitos por Módulo, Cobertura de Código)
- Métricas de Processo (ex.: Tempo Médio para Correção de Defeitos, Taxa de Retrabalho)
- Métricas de Projeto (ex.: Aderência ao Cronograma, Custo Estimado vs. Custo Real)
- Métricas de Recursos (ex.: Horas de trabalho da equipe)
- Todas têm impacto semelhante
- Nenhuma das categorias citadas tem impacto relevante

14. Os objetivos das métricas utilizadas em sua equipe são claramente definidos e documentados?

Marcar apenas uma oval.

- Sim, sempre
- Frequentemente
- Ocasionalmente
- Raramente
- Nunca

15. A organização em que você trabalha revisa e ajusta as métricas de qualidade ao longo do tempo para refletir mudanças nos objetivos do projeto?

Marcar apenas uma oval.

- Sim, regularmente
- Sim, com base em mudanças no projeto
- Apenas quando há problemas identificados
- Raramente
- Nunca

16. Qual o impacto que você atribui às métricas de qualidade na redução de defeitos do software?

Marcar apenas uma oval.

- Alto impacto
- Impacto moderado
- Baixo impacto
- Nenhum impacto
- Não sei responder

17. Sua equipe utiliza alguma metodologia de desenvolvimento específica (Ex: Scrum, Kanban, XP)? Se sim, poderia descrever qual(is) metodologia(s) são adotadas e como elas são aplicadas na prática no seu projeto?

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários