

Benchmarking para Desenvolvimento de Software: Um Estudo Bibliográfico

Benchmarking for Software Development: A Bibliographic Study
Benchmarking para el desarrollo de software: Un estudio bibliográfico

Rafael Gross¹

rafael.gross@cps.sp.gov.br

Antonio Cesar Galhardi²

antonio.galhardi@cps.sp.gov.br

1 – Faculdade de Tecnologia Deputado Ary Fossen | Fatec Jundiaí

2 – Unidade de Pós-graduação Ensino e Pesquisa - CEETEPS

Resumo:

O presente estudo aborda o uso de benchmarking no desenvolvimento de software, com foco em sua aplicação para elevar a qualidade e produtividade em projetos de sistemas logísticos. Por meio de uma revisão sistemática da literatura, foram analisados conceitos teóricos sobre benchmarking, integração com métodos de gestão de riscos (FMEA) e modelos de qualidade de software (ex.: ISO/IEC 25010). A pesquisa destaca que a falta de métricas padronizadas e dados históricos confiáveis limita a eficácia de benchmarks técnicos, levando a decisões mal fundamentadas sobre arquitetura de sistemas e alocação de recursos. Além disso, a combinação de benchmarking com FMEA revela-se uma estratégia promissora para mitigar falhas em fases iniciais de desenvolvimento, reduzindo retrabalho e custos operacionais. Como resultado, propõe-se uma metodologia para integrar benchmarks técnicos a práticas de gestão de riscos, validada por meio de análise crítica da literatura. As considerações finais indicam que o benchmarking, quando aliado a frameworks adaptáveis e dados históricos confiáveis, contribui significativamente para a melhoria contínua em projetos de software.

Palavras-chave: Benchmarking; Desenvolvimento de software; FMEA; Qualidade de software.

Abstract:

This study addresses the use of benchmarking in software development, focusing on its application to improve quality and productivity in logistics systems projects. Through a systematic literature review, theoretical concepts about benchmarking, integration with risk management methods (FMEA), and software quality models (e.g., ISO/IEC 25010) were analyzed. The research highlights that the lack of standardized metrics and reliable historical data limits the effectiveness of technical benchmarks, leading to poorly grounded decisions on system architecture and resource allocation. Additionally, combining benchmarking with FMEA proves to be a promising strategy to mitigate early-stage development failures, reducing rework and operational costs. As a result, a methodology is proposed to integrate technical benchmarks with risk management practices, validated through critical literature analysis. The findings indicate that benchmarking, when supported by adaptable frameworks and reliable historical data, significantly contributes to continuous improvement in software projects.

Keywords: Benchmarking; Software development; FMEA (Failure Mode and Effects Analysis); Software quality.

Recebido

Received

Recibido

Junho, 2026

Junho, 2026

Junho, 2026

Aceito

Accepted

Aceptado

Setembro, 2026

September, 2026

Septiembre, 2026

Publicado

Published

Publicado

Out./Nov. 2026

Oct./Nov. 2026

Oct./Nov. 2026

<https://git.fateczl.edu.br>

e_ISSN

2965-3339

DOI

10.5281/zenodo.23018178

São Paulo

v. 5 | n. 1

v. 5 | i. 1

e51550

Outubro-Dezembro

Outubro-Dezembro

Outubro-Dezembro

2026



Resumen:

Este estudio aborda el uso del benchmarking en el desarrollo de software, centrándose en su aplicación para mejorar la calidad y productividad en proyectos de sistemas logísticos. A través de una revisión sistemática de la literatura, se analizaron conceptos teóricos sobre benchmarking, integración con métodos de gestión de riesgos (FMEA) y modelos de calidad del software (por ejemplo, ISO/IEC 25010). La investigación destaca que la falta de métricas estandarizadas y datos históricos fiables limita la efectividad de los benchmarks técnicos, lo que conduce a decisiones poco fundamentadas sobre la arquitectura del sistema y la asignación de recursos. Además, combinar benchmarking con FMEA resulta ser una estrategia prometedora para mitigar fallos de desarrollo en etapas iniciales, reduciendo los costes de rework y operativos. Como resultado, se propone una metodología para integrar benchmarks técnicos con prácticas de gestión de riesgos, validadas mediante análisis crítico de la literatura. Los hallazgos indican que el benchmarking, cuando está respaldado por marcos adaptables y datos históricos fiables, contribuye significativamente a la mejora continua en proyectos de software.

Palabras clave: *Benchmarking; Desarrollo de software; FMEA (Análisis de Modos de Fallo y Efectos); Calidad del software.*

1 INTRODUÇÃO

Nos últimos anos, o desenvolvimento de software tornou-se um componente crítico para a eficiência operacional em setores estratégicos, como logística, saúde e indústria. No entanto, a crescente complexidade dos sistemas, aliada à entregas rápidas e produtos livres de falhas, expõe lacunas na qualidade e produtividade de processos de desenvolvimento. Apesar da ampla adoção de metodologias ágeis e ferramentas de automação, muitas organizações ainda enfrentam desafios como custos elevados de manutenção, falhas técnicas recorrentes e dificuldade em mensurar desempenho de forma objetiva (Bogan; English, 1994; Pressman, 2016).

O trabalho aborda como problemática em uma literatura previa da falta de métricas padronizadas para avaliar qualidade de software limita a capacidade de identificar melhorias processuais e técnicas. Estudos como os de Maxwell e Forselius (2000) revelam que benchmarks inadequados ou ausentes levam a decisões mal fundamentadas sobre arquitetura de sistemas e alocação de recursos. Além disso, a aplicação de benchmarks técnicos em projetos críticos carece de frameworks adaptáveis e dados históricos confiáveis para validar práticas de desenvolvimento (Pai et al., 2015; Kouznev et al., 2020). A dependência de estimativas subjetivas, como horas de trabalho, também contribui para imprecisões em planejamento e execução, conforme apontado por Czarnacka-Chrobot (2009). Essa problemática reforça a necessidade de abordagens estruturadas para superar barreiras na aplicação de benchmarking.

O objetivo geral deste trabalho é analisar como o benchmarking pode elevar a qualidade e produtividade no desenvolvimento de software. Para alcançar esse propósito, são definidos os seguintes objetivos específicos: 1. Avaliar ferramentas e frameworks de benchmarking técnico aplicáveis a softwares de gestão. 2. Validar a eficácia da abordagem proposta por meio de análise crítica da literatura e síntese teórica.

A pesquisa foi desenvolvida por meio de um estudo bibliográfico, com base em artigos científicos, livros e relatórios técnicos. A coleta de dados foi realizada em bases como *IEEE Xplore*, *ScienceDirect* e *Scopus*, priorizando estudos publicados entre 2010 e 2024 que abordem benchmarking e modelos de qualidade de software (ex.: ISO/IEC 25010).

2 FUNDAMENTAÇÃO TEÓRICA

A revisão bibliográfica sobre benchmarking no desenvolvimento de software revelou que esse conceito é amplamente aplicado para elevar a qualidade e produtividade em projetos, especialmente quando integrado a modelos de qualidade como ISO/IEC 25010. A seguir, são apresentados os principais tópicos

abordados, com destaque para uma síntese crítica dos estudos mais relevantes e suas contribuições para o campo.

2.1 Conceito e Importância do Benchmarking

O benchmarking é definido como um processo contínuo de comparação de produtos, serviços e práticas com líderes do setor, visando identificar melhores práticas e promover inovações (Bogan; English, 1994). No contexto do desenvolvimento de software, essa abordagem permite avaliar métricas técnicas como desempenho, segurança e escalabilidade e os processos organizacionais como produtividade de equipes para melhorar a qualidade do produto final (Maxwell; Forselius, 2000).

Segundo Czarnacka-Chrobot (2009), o benchmarking é importante para superar desafios como estimativas imprecisas de esforço, que levam a atrasos e estouros orçamentários em projetos. A autora enfatiza a importância de dados históricos de projetos similares para estimativas confiáveis, destacando que a falta de métricas objetivas ainda é um obstáculo persistente na área.

2.2 Aplicações do Benchmarking em Software

O benchmarking é aplicado em diferentes aspectos do desenvolvimento de software, como: métricas de produtividade — estudos como o de Maxwell e Forselius (2000) propõem equações para medir produtividade em projetos de software, considerando fatores como setor econômico e complexidade técnica; e benchmarking em projetos críticos — Pai et al. (2015) aplicaram Análise de Envoltória de Dados (DEA) para identificar projetos "*best practice*" em organizações certificadas *CMMI Level 5*, apontando um potencial de melhoria de 50% na produtividade.

Essas aplicações alinham-se ao primeiro objetivo específico do trabalho: avaliar ferramentas e frameworks de benchmarking técnico aplicáveis a softwares de gestão.

2.3 Modelos de Qualidade de Software

A qualidade de software é frequentemente avaliada por meio de padrões como ISO/IEC 25010, que define atributos como confiabilidade, usabilidade e manutenibilidade (Pressman, 2016). Benchmarks técnicos validam esses critérios, garantindo que o produto final atenda às expectativas do usuário e do mercado.

McCall et al. (1977) propuseram uma hierarquia de fatores de qualidade, dividindo-os em: fatores de produto (correção, confiabilidade, eficiência) e critérios de qualidade (métricas objetivas para avaliar cada fator, por exemplo, tempo médio entre falhas para confiabilidade).

Esses modelos de qualidade apoiam o segundo objetivo específico do estudo: validar a eficácia da abordagem proposta por meio de análise crítica da literatura e síntese teórica.

2.4 Desafios e Perspectivas Futuras

Apesar dos benefícios, desafios persistem na aplicação de benchmarking: qualidade dos dados — Bosu e MacDonell (2019) revelam que problemas como ruídos e incompletude em conjuntos de dados afetam a precisão de benchmarks técnicos; automação de benchmarks — ferramentas como *GraphGallery* (Li et al., 2021) ilustram avanços na automação de benchmarks para redes neurais gráficas (GNNs), mas a integração com análise preditiva ainda é limitada; e tecnologias emergentes — a aplicação de benchmarking em IA, blockchain e sistemas distribuídos requer frameworks adaptáveis, conforme discutido por Kouznev et al. (2020).

3 PROCEDIMENTOS METODOLÓGICOS

A presente pesquisa foi desenvolvida por meio de um estudo bibliográfico, com foco na análise de estudos científicos e técnicos sobre o uso de benchmarking no desenvolvimento de software. A metodologia adotada seguiu etapas estruturadas para garantir a abrangência, relevância e qualidade das fontes consultadas, conforme orientações da ABNT NBR 6023 (2018b).

A investigação teve caráter exploratório descritivo, baseando-se na coleta e análise crítica de referências bibliográficas em bases de dados científicas, como *IEEE Xplore*, *ScienceDirect*, *Scopus* e *SpringerLink*. Os critérios de seleção das fontes incluíram publicações entre 2010 e 2024, priorizando estudos recentes e atualizados, além de artigos, livros e relatórios técnicos que abordem benchmarking e modelos de qualidade de software. Foram priorizadas fontes com aplicações práticas em vários setores, alinhados ao contexto do trabalho.

A coleta de dados foi realizada por meio de buscas combinadas com palavras-chave como: *Benchmarking software development*, *software quality*, *Software productivity metrics* e *ISO/IEC 25010 benchmarking*. As referências foram organizadas em um banco de dados utilizando o software *Mendeley*, com filtros aplicados para eliminar duplicatas e artigos fora do escopo. A análise dos conteúdos seguiu duas vertentes: análise qualitativa, identificando conceitos-chave e modelos teóricos propostos nas fontes, e análise quantitativa, comparando métricas técnicas e estatísticas de benchmarks.

O estudo foi estruturado em três etapas principais: levantamento bibliográfico, síntese teórica e discussão crítica. Na primeira etapa, foram realizadas as buscas de referências nas bases mencionadas, aplicando critérios de seleção predefinidos. Na segunda, os temas foram organizados em categorias e relevância para o trabalho. Na terceira etapa, foram avaliadas as lacunas na literatura, como a falta de dados históricos confiáveis para aplicação do método em desenvolvimento de software (Bosu; MacDonell, 2019) e a necessidade de benchmarks padronizados para tecnologias emergentes (Kouznev et al., 2020).

Como a pesquisa é baseada em um estudo bibliográfico, algumas limitações foram observadas, como o viés de amostragem — a seleção de artigos pode não abranger todas as perspectivas relevantes, especialmente em línguas não inglesas — e a atualização de dados — algumas fontes, embora relevantes, datam de anos anteriores a 2010. Para mitigar essas limitações, foi priorizado o uso de estudos com alto índice de citações e publicações em eventos e periódicos de reconhecimento internacional, como os *proceedings* do ICSE (*International Conference on Software Engineering*).

4 RESULTADOS E DISCUSSÕES

O estudo bibliográfico permitiu identificar tendências, lacunas e contribuições significativas sobre o uso de benchmarking no desenvolvimento de software, com destaque para sua aplicação em setores estratégicos como logística, saúde e indústria. Os resultados foram organizados em três eixos principais: contribuições teóricas, desafios identificados e implicações para práticas futuras, conforme detalhado a seguir.

4.1 Contribuições Teóricas

A análise revelou que o benchmarking é reconhecido como ferramenta estratégica para elevar a qualidade e produtividade em projetos de software. Autores como Bogan e English (1994) e Maxwell e Forselius (2000) destacam que a comparação com melhores práticas do setor permite identificar gargalos em processos, otimizar alocação de recursos e reduzir custos operacionais. Estudos como os de Pai et al. (2015) reforçam a aplicação de benchmarks para mensurar produtividade em organizações certificadas *CMMI Level 5*, apontando um potencial de melhoria de 50% na eficiência de projetos.

Além disso, a literatura indica que modelos de qualidade, como ISO/IEC 25010, são fundamentais para validar benchmarks técnicos. Pressman (2016) enfatiza que atributos como confiabilidade, usabilidade e manutenibilidade devem ser mensurados por meio de métricas objetivas, garantindo que o produto atenda às expectativas do mercado. A hierarquia de fatores de qualidade proposta por McCall et al. (1977) também foi reiterada como base para avaliar critérios como eficiência e correção em sistemas complexos.

Essas contribuições teóricas alinham-se ao objetivo específico de avaliar ferramentas e frameworks de benchmarking técnico aplicáveis a softwares de gestão, destacando a relevância de métricas padronizadas para validar arquiteturas e processos.

4.2 Desafios Identificados

Apesar dos benefícios, a pesquisa evidenciou obstáculos persistentes na aplicação de benchmarking. Um dos principais desafios é a qualidade dos dados históricos, como destacado por Bosu e MacDonell (2019). Problemas como

ruídos, inconsistências e incompletude em conjuntos de dados afetam a precisão de estimativas de esforço e previsões de defeitos. Além disso, a falta de padronização em benchmarks técnicos limita sua utilidade, especialmente em tecnologias emergentes, conforme discutido por Kouznev et al. (2020).

Outro ponto crítico é a resistência cultural em organizações para adotar práticas de benchmarking de forma contínua. Rodrigues e Filho (2023) observam que muitas empresas ainda dependem de métricas subjetivas (horas de trabalho) em vez de parâmetros objetivos, como tamanho do software ou complexidade funcional. Essa abordagem inadequada contribui para estouros orçamentários e atrasos em projetos, corroborando os achados de Pai et al. (2015) em estudos com organizações *CMMI Level 5*.

4.3 Implicações para Práticas Futuras

Os resultados indicam que a democratização de benchmarks padronizados e a integração com metodologias ágeis são diretrizes prioritárias para o setor. A literatura sugere que frameworks adaptáveis, como os propostos por Kouznev et al. (2020), podem facilitar a aplicação de benchmarks em tecnologias disruptivas, como inteligência artificial e sistemas distribuídos. Além disso, a automação de processos, exemplificada por plataformas como *GraphGallery* (Li et al., 2021), demonstra potencial para reduzir o tempo de análise e aumentar a reprodutibilidade de resultados.

Quanto ao segundo objetivo específico — validar a eficácia da abordagem proposta por meio de análise crítica da literatura —, os achados corroboram a hipótese de que benchmarks técnicos integrados a modelos de qualidade ISO/IEC 25010 contribuem para a melhoria contínua em projetos de software.

4.4 Limitações da Amostragem

Como a pesquisa foi baseada em estudo bibliográfico, algumas limitações foram observadas. O viés de amostragem é uma delas, já que a seleção de artigos priorizou fontes em língua inglesa e com alto índice de citações, possivelmente negligenciando perspectivas regionais ou menos difundidas. Ademais, alguns estudos consultados datam de anos anteriores a 2010, o que pode limitar a atualização de certos conceitos em cenários tecnológicos rápidos. Para mitigar essas questões, priorizou-se o uso de referências de eventos internacionais e periódicos de alto impacto, como os indexados no *IEEE Xplore* e *ScienceDirect*.

5 CONSIDERAÇÕES FINAIS

A presente pesquisa teve como objetivo analisar como o benchmarking pode elevar a qualidade e produtividade no desenvolvimento de software, com foco em sua aplicação em projetos de sistemas. O estudo bibliográfico revelou que o benchmarking é uma ferramenta estratégica para identificar melhores práticas,

otimizar processos e reduzir custos operacionais, especialmente quando integrado a modelos de qualidade como ISO/IEC 25010.

Quanto ao problema de pesquisa inicial — a falta de métricas padronizadas e dados históricos confiáveis limitando a eficácia de benchmarks técnicos —, os resultados confirmam essa lacuna, corroborando estudos como os de Bosu e MacDonell (2019) e Kouznev et al. (2020). A dependência de estimativas subjetivas como horas de trabalho foi identificada como um obstáculo persistente, levando a decisões mal fundamentadas sobre arquitetura de sistemas e alocação de recursos.

Em relação aos objetivos específicos: avaliação de ferramentas e frameworks de benchmarking técnico — a análise crítica da literatura evidenciou a necessidade de benchmarks adaptáveis para tecnologias emergentes, como inteligência artificial e blockchain, destacando a ausência de padronização como um desafio central. Estudos como os de Maxwell e Forselius (2000) e Pai et al. (2015) reforçam a importância de métricas objetivas para validar práticas de desenvolvimento; validação da eficácia da abordagem proposta — a integração de benchmarks técnicos a modelos de qualidade como ISO/IEC 25010 demonstra potencial para garantir atributos críticos, como confiabilidade e escalabilidade, conforme discutido por Pressman (2016) e McCall et al. (1977).

A metodologia, baseada em estudo bibliográfico, demonstrou-se adequada para mapear lacunas teóricas e propor diretrizes práticas. No entanto, limitações como o viés de amostragem (priorização de fontes em língua inglesa e com alto índice de citações) e a atualização de dados (algumas fontes datam de anos anteriores a 2010) foram reconhecidas. Para trabalhos futuros, recomenda-se a realização de estudos de caso em organizações reais, com coleta de dados quantitativos sobre a eficácia de benchmarks em ambientes diversos com foco em desenvolvimento. Além disso, a criação de frameworks padronizados para tecnologias emergentes pode ampliar a aplicabilidade do benchmarking em cenários dinâmicos, como a indústria 4.0.

Como contribuição prática, este trabalho destaca a necessidade de: 1. Democratização de benchmarks técnicos — promover o uso de ferramentas acessíveis e métricas padronizadas para organizações de todos os portes; 2. Integração com metodologias ágeis — adaptar benchmarks para ciclos ágeis de desenvolvimento, alinhando-os a práticas como *Scrum* e *DevOps*; 3. Formação profissional — capacitar equipes em técnicas de benchmarking e análise de riscos, como parte de programas de formação em engenharia de software.

Em síntese, o benchmarking demonstra-se importante para a evolução do setor de desenvolvimento de software, mas sua eficácia depende de investimentos em dados históricos confiáveis, automação de processos e adaptação a tendências tecnológicas. Este estudo contribui ao mapear caminhos para superar barreiras atuais e explorar oportunidades futuras na área, alinhando-se ao escopo bibliográfico definido.

REFERÊNCIAS

- BOGAN, C. E.; ENGLISH, M. J. Benchmarking for Best Practices: Winning Through Innovative Adaptation. New York: McGraw-Hill, 1994.
- BOSU, M. F.; MACDONELL, S. G. Data Quality in Empirical Software Engineering: A Systematic Review. Journal of Data and Information Quality (JDIQ), v. 11, n. 4, p. 1-38, 2019.
- CZARNACKA-CHROBOT, B. The Role of Benchmarking Data in the Software Development and Enhancement Projects Effort Planning. Frontiers in Artificial Intelligence and Applications, v. 199, p. 106-127, 2009.
- ISO/IEC. ISO/IEC 25010: Sistemas e software — Modelos de qualidade. Genebra: ISO, 2017.
- KOUZNEV, S.; LANGE, K.-D.; KISTOWSKI, J. von. Systems Benchmarking for Scientists and Engineers. Springer, 2020.
- MAXWELL, K. D.; FORSELIUS, P. Benchmarking software development productivity. IEEE Software, v. 17, p. 80-88, 2000.
- PRESSMAN, R. S. Engenharia de Software: uma abordagem profissional. McGraw-Hill, 2016.
- PAI, D. R.; SUBRAMANIAN, G. H.; PENDHARKAR, P. C. Benchmarking software development productivity of CMMI level 5 projects. Information Technology and Management, v. 16, p. 235-251, 2015.
- RODRIGUES, C. A.; FILHO, R. R. R. Aplicação da ferramenta benchmarking em performance operacional. Revista Interface Tecnológica, v. 20, n. 1, p. 1-12, 2023.

Os conteúdos expressos no trabalho, bem como sua revisão ortográfica e adequação às normas ABNT são de inteira responsabilidade dos autores.

Declara-se pelos autores que durante a preparação deste trabalho foi utilizada a ferramenta QWEN IA para correção textual e abstract. Após utilizar esta ferramenta/serviço, os autores editaram e revisaram o conteúdo conforme necessário e assumem total responsabilidade pelo conteúdo da publicação.