

AN ANALYSIS OF THE USE OF ARTIFICIAL INTELLIGENCE IN SOFTWARE DEVELOPMENT

João Paulo de Souza Rodrigues - Faculdade de Computação / Universidade Federal do Pará - <https://orcid.org/0009-0006-2906-9847>

Sandro Ronaldo Bezerra Oliveira - Faculdade de Computação / Universidade Federal do Pará - <https://orcid.org/0000-0002-8929-5145>

ABSTRACT: With the growth of new artificial intelligence tools capable of assisting in the construction of software in increasingly faster and optimized ways, there is a need to analyze the impacts that these tools have on the development of new applications. This article aims to conduct a literature review to evaluate the effects of adopting artificial intelligence tools in software development. As a result, this work provides an overview of the impacts and challenges encountered in the adoption of these tools in the field of software engineering.

Keywords: Software development, Artificial intelligence, Impacts, Review

UMA ANÁLISE SOBRE O USO DE INTELIGÊNCIA ARTIFICIAL NO DESENVOLVIMENTO DE SOFTWARE

RESUMO: Com o crescimento de novas ferramentas de inteligência artificial capazes de auxiliar na construção de software de forma cada vez mais rápida e otimizada, há uma necessidade de analisar os impactos que estas ferramentas têm sobre o desenvolvimento de novas aplicações. Este artigo tem como objetivo realizar uma revisão da literatura para avaliar os efeitos da adoção de ferramentas de inteligência artificial no desenvolvimento de software. Como resultado, este trabalho apresenta um panorama geral sobre os impactos e os desafios encontrados na adoção dessas ferramentas na área de engenharia de software.

Palavras-chave: Desenvolvimento de software, Inteligência artificial, Impactos, Revisão

Agradecimentos: Este trabalho pertence ao Projeto SPIDER (<https://projeto-spider.github.io/>)

1. INTRODUÇÃO

Durante a pandemia de COVID-19, o mundo vivenciou uma rápida transformação digital. A necessidade de inovação de vários setores da economia impulsionou o crescimento da indústria de software no Brasil. Segundo dados da Associação Brasileira das Empresas de Software (ABES, 2022) a indústria de tecnologia cresceu 17,4% durante o ano de 2021 ficando acima da média mundial.

Com o aumento na demanda por novas soluções de tecnologia, a qualidade de software tornou-se uma preocupação ainda maior para as empresas de desenvolvimento. Nos Estados Unidos, os custos causados pela má qualidade do software em 2022 foram estimados em US\$ 2,41 Tri. Além dos gastos com a baixa qualidade de software, os gastos com a dívida técnica cresceram desde a última avaliação em 2020, sendo estimados em US\$ 1,52 Tri (KRASNER, 2023). De acordo com o relatório publicado em 2018 pela empresa de tecnologia STRIPE (2018), em média, os desenvolvedores gastaram 13,5 horas com dívidas técnicas e mais 3,8 horas na remoção de códigos ruins.

Além dos custos da baixa qualidade de software e do tempo gasto com dívidas técnicas, um dos principais motivadores para a adoção das ferramentas de IA é a falta de profissionais de tecnologia. Segundo dados da Associação Brasileira de Tecnologia da Informação e Comunicação e de Tecnologias Digitais (BRASSCOM, 2021) até 2025 o Brasil pode alcançar o déficit de 530 mil profissionais de tecnologia.

Os avanços das técnicas de inteligência artificial (IA) na engenharia de software possibilitaram a construção de ferramentas autonomizadas para a fase de desenvolvimento. Com o crescimento na adoção de ferramentas de inteligência artificial pelas empresas de tecnologia faz-se necessário avaliar quais os efeitos da adoção dessas ferramentas para o desenvolvimento de software. Este artigo tem como objetivo avaliar os impactos e os avanços alcançados pelas ferramentas de inteligência artificial no desenvolvimento de software por meio de uma revisão da literatura.

Este trabalho foi estruturado em 4 seções além da seção de introdução. Na Seção 2 será apresentada a fundamentação teórica deste trabalho. Em seguida, será apresentada na Seção 3 a metodologia da pesquisa que conduziu este trabalho bem como a seleção dos artigos e os resultados encontrados. Na Seção 4 serão discutidos os resultados da pesquisa e seus impactos. Por fim, na Seção 5 será feita a conclusão do artigo, dispendo sobre as contribuições feitas por este trabalho, as limitações encontradas e trabalhos futuros.

2. FUNDAMENTAÇÃO TEORICA

Nesta seção será apresentada a fundamentação teórica do artigo que se estrutura em três partes: desenvolvimento de software, qualidade de software e inteligência artificial.

2.1. Desenvolvimento de Software

Entre as décadas de 1960 e 1970, o aumento na demanda por novos programas de computador gerou uma crise na indústria de software. A ausência de metodologias e de técnicas padronizadas de desenvolvimento resultava frequentemente em alto custo de desenvolvimento, baixa qualidade de software e falhas na execução do projeto (WAZLAWICK, 2019).

A conferência sobre engenharia de software realizada pela OTAN (Organização do Tratado do Atlântico Norte) na cidade de Garmisch na Alemanha em 1968, resultou em um esforço coletivo para estabelecer padrões mais sólidos no desenvolvimento de software (VALENTE, 2021).

A partir da conferência de Gamisch, o processo de desenvolvimento passou a adotar padrões e princípios da área de engenharia com o objetivo de melhorar a qualidade, a produtividade e a manutenção do software.

Conforme definido pela norma ISO/IEC/IEEE 24765 o desenvolvimento de software é o processo de transformar as necessidades do usuário em um produto de software. Esse processo engloba várias atividades, como levantamento e análise de requisitos, projeto, codificação, testes e manutenção do software.

A fase de codificação tem como objetivo implementar os requisitos em uma linguagem de programação. O código desenvolvido deve seguir critérios rígidos de desenvolvimento para a garantia da qualidade e conformidade com os requisitos desenvolvidos.

A ausência de padrões no desenvolvimento e de boas práticas de programação tem um efeito direto no tempo de vida do software. A ausência na adoção dessas boas práticas pode ocasionar no aumento de dívidas técnicas, perda de produtividade das equipes de desenvolvimento, atrasos na entrega e o aumento da complexidade do software desenvolvido conforme é descrito no livro Clean Code (MARTIN, 2019).

2.2. Qualidade de Software

A ISO/IEC/TR24772-1 define a qualidade de software como o grau em que o software implementa os requisitos descritos pela sua especificação e o grau em que as características de um produto de software satisfazem os seus requisitos.

Entretanto, durante anos, a indústria de software priorizou a velocidade no desenvolvimento de novos recursos em detrimento da qualidade de software. Os testes de segurança eram feitos apenas após a conclusão do projeto, os erros encontrados eram corrigidos e adicionados em um novo *patch* de segurança (WILLIAMS, 2019). Em um estudo publicado por BOEHM e PAPACCIO (1988), os autores apontaram uma tendência de crescimento nos custos de manutenção de software à medida em que um erro avançava pelas fases do ciclo de desenvolvimento. Esse estudo reforçou a necessidade de adotar novas abordagens de desenvolvimento com o foco na qualidade do software.

Para PAULK (1995), a qualidade do produto de software é diretamente ligada à qualidade dos processos implementados durante o ciclo de desenvolvimento. A adoção de modelos de qualidade durante as fases do ciclo de desenvolvimento foi essencial para a melhoria da qualidade do software desenvolvido.

Segundo a norma ISO/IEC 25010, os atributos de qualidade do produto de software são divididos em 8 classes: 1. Funcionalidade, 2. Eficiência de performance, 3. Compatibilidade, 4. Usabilidade, 5. Confiabilidade, 6. Segurança, 7. Manutenibilidade, 8. Portabilidade. Esses critérios são utilizados para guiar o processo de implementação de um sistema e são adotados pelos ciclos de desenvolvimento de software.

2.3. Inteligência Artificial (IA)

Nos últimos anos, as aplicações de inteligência artificial popularizaram-se pelo mundo. Os avanços obtidos nas áreas processamento de dados e nos métodos de treinamento permitiram a ampliação das áreas de atuação de IA. Segundo o levantamento da IBM (2022), 41% das empresas brasileiras implementam ativamente ferramenta de IA em seus negócios. Entre os fatores que impulsionam o crescimento na adoção dessas ferramentas estão a redução de custos, automatização de tarefas e a falta de profissionais no mercado de trabalho.

John McCarthy (2007) define o termo inteligência artificial como "a ciência e a engenharia de fazer máquinas inteligentes, especialmente programas de computador

inteligentes". Em 1950, Turing inicia os primeiros debates sobre a área de inteligência artificial. Em seu artigo "*Computing Machinery and Intelligence*", Turing discute sobre a possibilidade de as máquinas agirem de maneira inteligente, sendo capazes de se passarem por humanos como em um jogo de imitação. Desde então, várias pesquisas foram desenvolvidas visando tornar a construção desses sistemas algo possível.

As pesquisas sobre IA podem ser divididas em quatro categorias: 1. Sistemas que pensam como humanos, 2. Sistemas que agem como humanos, 3. Sistemas que pensam racionalmente, 4. Sistemas que agem racionalmente (RUSSELL; NORVIG, 2010). Os estudos publicados ao longo do tempo permitiram a aplicação da IA em áreas distintas do conhecimento, como na área da saúde (sistema de assistência médica), atendimento ao cliente (*chatbot*) e carros autônomos.

3. REVISÃO DA LITERATURA

Este artigo tem como objetivo realizar uma pesquisa exploratória em uma base acadêmica de dados com o objetivo de avaliar os impactos das ferramentas de IA na fase de desenvolvimento de software. O objetivo da pesquisa exploratória é reunir dados e informações sobre um tema específico para definir e delimitar sua área de estudo. Isso envolve mapear as condições que caracterizam esse tema, conforme apontado por Severino (2016).

A metodologia deste trabalho foi estruturada em 4 partes, como apresentado na Figura 1. A primeira fase constituiu-se em desenvolver uma *string* de busca para aplicar em uma base de dados. A segunda etapa consiste na seleção de uma base de dados científica para as buscas. Na terceira fase da metodologia foram criados os critérios de inclusão e exclusão para a pesquisa e por fim é feita a extração de dados sobre o assunto.

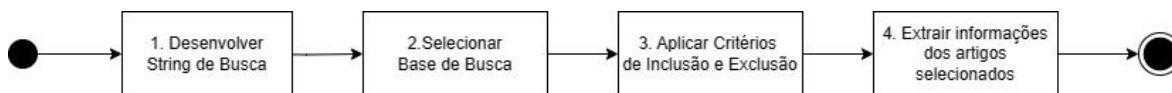


Figura 1 – Processo de Revisão da Literatura

Fonte: Elaboração própria (2023).

3.1. Estratégia de Busca

Nessa etapa conduziu-se uma busca na base acadêmica de dados do Google Scholar utilizando como parâmetros de pesquisa uma *string* de busca filtrada por título apresentada no Quadro 1. A busca ocorreu entre os meses de junho e agosto de 2023.

Quadro 1 – *String* de busca selecionada

String de Busca
allintitle: "software development", "artificial intelligence"OR "AI-Assisted"

Fonte: Elaboração própria (2023).

3.2. Seleção dos Trabalhos

Nesta seção são apresentados detalhadamente os critérios inclusão (CI) e exclusão (CE) que guiaram a seleção dos trabalhos acadêmicos. Os critérios de CI e CE podem ser observados no Quadro 2.

Quadro 2 - Critério de seleção dos estudo primários

ID	Critério de Inclusão (CI) ou Critério de Exclusão (CE)
CI1	Publicações que descrevem sobre a aplicação de ferramentas de IA no desenvolvimento de software
CI2	Publicações que descrevem sobre o impacto das ferramentas artificiais na fase de desenvolvimento de software
CE1	Publicações anteriores a 2015
CE2	Publicações que não estejam escritos em inglês ou português
CE3	Estudos não disponíveis para download abertamente ou através do IP institucional dos pesquisadores
CE4	Estudos como relatórios de workshops, pôsteres, apresentações, palestras de palestrantes, livros, teses e dissertações.

Fonte: Elaboração própria (2023).

3.3. Resultados

Na busca inicial no Google Acadêmico, foram identificados 78 artigos a partir dos termos de pesquisa estabelecidos. Em seguida, aplicaram-se critérios de inclusão e exclusão conforme descritos anteriormente. O Quadro 3 ilustra as métricas associadas a cada critério, resultando na seleção de apenas 5 artigos, os quais são detalhados no Quadro 4. Para garantir uma abordagem mais ampla do desenvolvimento, optou-se por excluir publicações tangenciais ao tema e aquelas focadas em aplicações específicas da indústria de software.

Quadro 3 - Seleção de estudos

Base de Dados	Artigos Retornados	CE1	CE2	CE3	CE4	CI1	CI2
Google Acadêmico	78	56	54	46	37	9	5

Fonte: Elaboração própria (2023).

Embora o tema seja relevante, a literatura acadêmica existente é predominantemente estrangeira, revelando uma lacuna significativa na produção nacional. Os artigos disponíveis concentram-se na utilização de ferramentas de inteligência artificial nas fases iniciais do processo de desenvolvimento, como na coleta e análise de requisitos, além de abordarem as etapas finais, como a implementação de testes.

Quadro 4 – Artigos selecionados

Título	Autores	Ano
Use of artificial intelligence in Software development lifecycle	Bhagyashree W. Sorte, Pooja P. Joshi, Vandana Jagtap	2015
How Artificial Intelligence Accelerates Software Development	Divyanshi Kothari	2019
Artificial intelligence for software development — the present and the challenges for the future	Lukasz Korzeniowski, Krzysztof Goczyła	2019
Development Trend of Computer Artificial Intelligence Technology Optimization Strategy in Software Development	Saidong Li	2021
Crossover of Artificial Intelligence and Software Development Lifecycle	Amna Raza Khan, Javeria Fatima, Sumaira Ahmed	2023

Fonte: Elaboração própria (2023).

3.4. *How Artificial Intelligence Accelerates Software Development*

O artigo tem como objetivo analisar as abordagens de IA sobre os processos de engenharia de software e entender os seus impactos no desenvolvimento de software. Os autores apresentam o ciclo desenvolvimento como sendo um processo longo e complexo dividido em múltiplas fases que deve resultar em um código viável e funcional no fim do projeto.

A aplicação de IA em cada uma das fases de desenvolvimento visa, sobretudo, diminuir o intervalo entre as fases de levantamento de requisitos e entrega do produto aumentando a eficiência da equipe de desenvolvimento e diminuindo o tempo de espera da entrega do projeto. Na fase de implementação, a autora descreve o uso da IA como sendo o segundo desenvolvedor na programação em par sendo capaz de auxiliar o ser humano na criação de funções, estruturas de código e automatização de tarefas.

Além de fatores como eficiência e desempenho das equipes, a qualidade do software desenvolvido tende a melhorar com a implementação de IA durante o ciclo de desenvolvimento de software reduzindo os custos de manutenção e riscos de erro.

3.5. *Use of artificial intelligence in Software development life cycle*

Para este trabalho os autores (KHAN; FATIMA; AHMED, 2023) realizaram uma revisão do estado da arte sobre o uso de Inteligência Artificial no ciclo de vida do processo de desenvolvimento de software (SDLC). A aplicação de ferramentas de IA na engenharia de software é constantemente impulsionada pela criação de novas metodologias e tecnologias com o objetivo de criar novos sistemas capazes de auxiliar e automatizar as atividades do ciclo de desenvolvimento de software.

Um dos casos bem-sucedidos é a adoção dessas ferramentas de IA para detecção de falhas e vulnerabilidades no código fonte. A adoção na fase de testes permitiu o aumento da segurança da aplicação e a redução de riscos no ambiente de produção. Na fase de geração de código, os autores mencionam o uso de métodos e ferramentas para programação automatizada que resultam na geração de código através de modelos matemáticos, textuais e UML.

Os códigos gerados por esses métodos seguem políticas e regras predefinidas implicando em desafios para a construção de software. Esses desafios encontrados são ultrapassados com o processo de geração automática de software orientada a agentes. A abordagem orientada a agentes permite a autonomia do SDA (*Software Development Agent*) para tomar decisões sobre como analisar, projetar e implementar o sistema sem a necessidade de uma intervenção humana.

Os autores concluem o trabalho ressaltando a importância da automação no processo de desenvolvimento de software majoritariamente manual, e a necessidade de novos estudos sobre os impactos da integração das áreas de engenharia de software e inteligência artificial.

3.6. *Development Trend of Computer Artificial Intelligence Technology Optimization Strategy in Software Development*

Este artigo tem como objetivo avaliar os efeitos da aplicação de inteligência artificial no desenvolvimento de software por meio de um experimento realizado com duas empresas de tecnologia, denominadas no artigo de 'A' e 'B'. O estudo separou dados de desempenho das equipes e da preferência dos consumidores finais. O grupo 'A' aplicou ferramentas de IA em seu ciclo de desenvolvimento, enquanto o grupo 'B' atuou como grupo de controle.

Durante o período de um ano foram realizadas quatro coletas de dados em intervalos de três meses para medir o desempenho das equipes. Após a entrega do produto

foram realizadas pesquisas de satisfação com duzentos clientes e avaliou o nível de satisfação obtido por cada produto.

Na primeira coleta de dados o grupo 'A' obteve 82% de eficiência em sua primeira avaliação e finalizou a coleta de dados com 97% apresentando um progresso nos níveis de eficiência registrados. O grupo de controle iniciou suas métricas de eficiência em 78%, as métricas de eficiência do grupo de controle não apresentaram um crescimento regular entre as análises variando entre os intervalos de 70-80%, encerrando o período de análises de eficiência com 72%. A análise comparativa desses dados apresentou um efeito positivo no uso de inteligência artificial no desenvolvimento do software aumentando a produtividade e desempenho das equipes que adotaram IA durante a fase de desenvolvimento de software.

Na pesquisa sobre o nível de satisfação dos usuários por cada produto. O nível de satisfação de cada cliente foi avaliado em uma escala com quatro níveis: muito satisfeito, satisfeito, regular e insatisfeito. O nível de aprovação foi calculado a partir da soma do número de consumidores que ficaram muito satisfeitos com o número de consumidores que ficaram apenas satisfeitos com o produto. Os dados apontaram um nível de aprovação maior dos clientes do grupo 'A', apresentando 65% de aprovação. No grupo 'B', o nível de aprovação dos entrevistados foi de 36%.

Os resultados das duas pesquisas apresentaram impactos positivos e relevantes sobre a implementação de inteligência artificial do desenvolvimento de software, sugerindo uma vantagem competitiva em termos de eficiência e satisfação do cliente para as empresas de tecnologia.

3.7. *Artificial intelligence for software development — the present and the challenges for the future*

Este trabalho teve como objetivo realizar uma revisão de literatura para avaliar os avanços e os desafios que a inteligência artificial possui na área de engenharia de software. A implementação de IA no processo de desenvolvimento de software, segundo os autores, justifica-se por três motivos: o processo tradicional de desenvolvimento é predominantemente manual, caro e propenso a erros; o uso de softwares de prateleira é custoso e exige constantes atualizações; e a falta de desenvolvedores no mercado dificulta no desenvolvimento de sistemas mais complexos.

As pesquisas na área de programação automatizada concentram-se em três abordagens: síntese de programas, indução de programas e uma abordagem híbrida dos dois métodos. Para cada abordagem os autores apresentam os pontos positivos e negativos de cada método. A ausência de ferramentas de IA capazes de gerar e manter o próprio código resulta em um desafio de gerar código legível para os humanos, para garantir a possibilidade de manutenção do código desenvolvido, o que para os autores é considerado uma limitação técnica no desenvolvimento dessas ferramentas.

Os autores classificam as ferramentas de IA em três níveis: *Full AI*, em que a IA é capaz de criar programas independentes; *Assisted AI*, em que a IA trabalha em conjunto com o desenvolvedor para criar programas de forma mais eficiente e segura; e *Supplementary AI*, em que a IA desempenha um papel secundário no processo de desenvolvimento de software, auxiliando nas melhorias de qualidade e no desempenho do software criado manualmente.

A ferramentas *Full AI* possuem o foco da maioria das pesquisas na área. Entre os desafios encontrados pelos pesquisadores estão a eficiência dos processos de aprendizado de máquina e precisão do código gerado por essas ferramentas.

A adoção da inteligência artificial no processo desenvolvimento de software começa a apresentar seus primeiros resultados. Apesar da incapacidade das ferramentas de

IA de construir e manter o código. A adoção dessas ferramentas permite auxiliar o desenvolvedor no processo de desenvolvimento melhorando a qualidade do código, verificando falhas e automatizando tarefas.

3.8. *Crossover of Artificial Intelligence and Software Development Lifecycle*

O uso de IA vem se expandindo em vários setores da indústria, bem como no setor de desenvolvimento de software. A ideia de existir uma IA capaz de construir e manter um software ainda está longe de ser viável, porém é uma área promissora.

Apesar dos avanços na engenharia de software, o ciclo de desenvolvimento continua sendo extremamente manual e propício a falhas durante o processo, possibilitando o desvio das metas de desenvolvimento bem como erros na implementação de requisitos. O uso da IA no desenvolvimento de software tem como objetivo preencher as lacunas presentes no processo e melhorar o ciclo de desenvolvimento. A redução e a remoção de erros, o aumento da produtividade das equipes de desenvolvimento e a redução dos custos de operação incentivam cada vez mais a adoção dessa ferramenta no ciclo de desenvolvimento.

Com o cenário cada vez mais competitivo, a adoção de ferramentas *low-code/no-code* vem crescendo na indústria de software. A adoção dessas ferramentas permite uma rápida customização de aplicações prontas para a produção sem a necessidade de uma equipe experiente na área software.

A interseção entre as áreas de engenharia de software e inteligência artificial possibilitou a ampliação de novas áreas de pesquisa, além de reduzir o tempo de desenvolvimento de novas aplicações por meio da reutilização de código pronto permitindo a redução de linhas de código no programa. Apesar das vantagens de implementação, há riscos que devem ser pontuados e avaliados no ciclo de desenvolvimento de software.

Os autores apontam a integração entre as ferramentas de *no-code* e inteligência artificial como uma revolução para o ciclo de desenvolvimento de software. Apesar disso, os autores destacam que os estudos sobre o assunto ainda são recentes e necessitam de mais pesquisas para serem validadas.

4. DISCUSSÃO

Os avanços nas pesquisas sobre inteligência artificial possibilitaram a implementação de técnicas de *Machine Learning*, *Deep Learning* e Processamento de Linguagem Natural em vários setores de atuação permitindo a rápida popularização da área.

Nos últimos anos, a indústria de software vem ampliando o uso de ferramentas de IA para a automatização de testes, detecção de erros e geração de código. Entre as principais ferramentas de desenvolvimento podemos citar o Visual Studio IntelliCode e Github Copilot.

Devida a complexidade de se construir essas ferramentas, as principais ferramentas de desenvolvimento com inteligência artificial são softwares proprietários. A adoção dentro do processo de implementação deve ser avaliada em termos de custo para a empresa.

A adoção de ferramentas de inteligência artificial durante o desenvolvimento de software permitiu a redução de erros no código, detecção de vulnerabilidades, e melhorias de desempenho nos algoritmos resultando em uma maior garantia de qualidade do software desenvolvido.

Além de fatores técnicos, a adoção dessas ferramentas possibilitou um aumento de produtividade por parte dos desenvolvedores envolvidos no projeto. A geração automática de estruturas de dados e a revisão automática do código diminuí os custos obtidos com

dividas técnicas e baixa qualidade do software escrito permitindo a melhoria das métricas de eficiência da equipe. Os resultados encontrados conciliam com os resultados apontados por (BARENKAMP; THOMAS; THOMAS, 2020) em seu livro “*Applications of AI in classical software engineering. AI Perspectives*”.

Apesar das ferramentas de IA terem como propósito a redução de tarefas feitas manualmente, o código gerado não é 100% passível de confiança, o que pode ocasionar em uma necessidade de intervenção do desenvolvedor para corrigir possíveis erros ou vulnerabilidades geradas e ainda não removidas das bases de treinamento da ferramenta (KORZENIOWSKI; GOCZYLA, 2019). A falha na geração de código por parte da IA tende a diminuir à medida que novos avanços na área de aprendizado e de desenvolvimento forem alcançados.

A ausência de avaliações quantitativas dos impactos da adoção de ferramentas de inteligência no desenvolvimento de software na literatura acadêmica dificultou a medição dos efeitos dessa adoção.

5. CONCLUSÃO

A adoção de inteligência artificial na engenharia de software vem promovendo uma evolução no campo de desenvolvimento de software. A adoção dessas ferramentas é estudada em várias fases do ciclo de desenvolvimento. Este artigo teve como objetivo realizar uma revisão de literatura para avaliar os impactos da adoção de ferramentas de inteligência artificial no desenvolvimento e implementação do software.

Os resultados encontrados apontaram impactos positivos quanto ao uso de ferramentas de inteligência artificial durante a fase de desenvolvimento, tendo como resultados a redução do tempo de desenvolvimento, o aumento da produtividade dos desenvolvedores e melhorias na qualidade do software.

As limitações da pesquisa incluem a restrição da pesquisa a uma única base de dados, limitando a visão geral dos estudos disponíveis. Além disso, a filtragem dos artigos apenas por título pode reduzir o número de estudos que discutem sobre o tema desta pesquisa.

Para os trabalhos futuros, este artigo propõe realizar uma análise quantitativa mais extensa, utilizando métricas específicas para medir os impactos das ferramentas de IA. Isso pode incluir estudos de caso detalhados em projetos reais e a comparação de desempenho antes e depois da adoção das ferramentas de IA.

Estes trabalhos representam uma oportunidade significativa para contribuições valiosas à literatura, permitindo a realização de estudos mais detalhados e específicos com o objetivo de aprofundar o conhecimento sobre as ferramentas de inteligência artificial.

REFERÊNCIAS BIBLIOGRÁFICAS

ABES. Mercado Brasileiro de Software: panorama e tendências, 2022.

BARENKAMP, M.; REBSTADT, J.; THOMAS, O. Applications of AI in classical software engineering. *AI Perspectives*, v. 2, n. 1, p. 1, 2020.

BOEHM, B. W.; PAPACCIO, P. N. Understanding and controlling software costs. *IEEE transactions on software engineering*, v. 14, n. 10, p. 1462-1477, 1988.

BRASSCOM. Estudo da Brasscom aponta demanda de 797 mil profissionais de tecnologia até 2025, 2021.

IBM. Estudo IBM: 41% das empresas no Brasil já implementaram ativamente inteligência artificial em seus negócios. IBM Comunica, 2022.

ISO/IEC/IEEE 24765 - Systems and software engineering — Vocabulary. Geneva, Switzerland: ISO/IEC, 2017.

ISO/IEC 25010 - Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, 2011.

ISO/IEC/TR24772-1 - Programming languages – Guidance to avoiding vulnerabilities in programming languages – Part 1: General, 2019.

KHAN, A.; FATIMA, J.; AHMED, S. Crossover of Artificial Intelligence and Software Development Lifecycle. International Journal of Computing and Related Technologies, 4(1), 35-42. 2023

KORZENIOWSKI, L.; GOCZYLA, K. Artificial intelligence for software development: the present and the challenges for the future Biuletyn Wojskowej Akademii Technicznej v68 n1-2019

KOTHARI, D. How artificial intelligence accelerates software development International Research Journal of Engineering and Technology (IRJET), v. 6, n. 8, p. 1392-1394, 2019.

KRASNER, H. Cost of Poor Software Quality in the U.S.: A 2022 Report. CISQ, Jan 2023.

LIU, S. Development Trend of Computer Artificial Intelligence Technology Optimization Strategy in Software Development Journal of Physics: Conference Series v2037 p12006-2021

MARTIN, R. C. Clean Code: A Handbook of Agile Software Craftsmanship. 1. ed. USA: Prentice Hall PTR, 2008.

MCCARTHY, J. et al. What is artificial intelligence. Stanford University, 2007.

PAULK, M. C. et al. The Capability Maturity Model: Guidelines for Improving the Software Process. USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

RUSSELL, S.; NORVIG, P. Artificial Intelligence: A Modern Approach, 2010.

Severino, A. J. Metodologia do trabalho científico Cortez editora, 2016.

SORTE, B. W.; JOSHI, P. P.; JAGTAP, V. Use of artificial intelligence in software development life cycle—a state of the art review International Journal of Advanced Engineering and Global Technology v3 n3 p398-403-2015

STRIPE. The Developer Coefficient, 2018.

TURING, A. Computing Machinery and Intelligence, 1950.

VALENTE, M.T Engenharia de software moderna Princípios e Práticas para Desenvolvimento de Software com Produtividade v.1 p24-2020

WAZLAWICK, Raul. Engenharia de software: conceitos e práticas. Elsevier Editora Ltda., 2019.

Williams, L. Secure Software Lifecycle Knowledge Area Issue 1.0. In A. Martin (Ed.), CyBOK – The Cyber Security Body of Knowledge (pp. 517-550), 2019.