Machine Learning Research
2024, Vol. 9, No. 2, pp. 39-47
https://doi.org/10.11648/j.mlr.20240902.12

Research Article

Generative Artificial Intelligence: Challenges and Opportunities for Systems Developers: A Systematic Mapping of Literature

Samira Santos Caduda, Anderson Santos Barroso*

Department of Computing, Tiradentes University, Aracaju-SE, Brazil

Abstract

Generative Artificial Intelligence tools have gained increasing prominence in recent years. However, the increasing use of these technologies and the functionalities they offer has sparked discussions about their impact and even raised concerns about the potential replacement of human work by automation carried out by machines. This study proposes a Systematic Literature Review to evaluate the opportunities and challenges that these technologies present to system developers in the current and future technological scenario. Aiming at state-of-the-art research to identify how Generative AIs are being applied in the context of software development and what are the latest trends and innovations in this field and how these innovations affect the opportunities and challenges for system developers. As a result, several studies were found that highlight how Generative AI has provided productivity and systems development optimized solutions in the industry, as well as promoting innovations. Studies also emphasize the need for a balance between the use of AI tools and development carried out by human participation, which must be mediated by common sense. Furthermore, the review will explore the ethical implications associated with the widespread adoption of AI technologies, addressing issues such as data privacy, decision-making transparency, and the responsibility of developers in ensuring that AI applications are used in a way that benefits society. The findings of this review will contribute to a better understanding of how generative AI is reshaping the software development landscape and provide insights for future research and development in this rapidly evolving field.

Keywords

Generative Artificial Intelligence, System Development, Systematic Mapping

1. Introduction

At the beginning of the 19th century, there were several protests against technological innovations in regarding the industrial revolution. Workers invaded and broke several machines, motivated by possibility of unemployment resulting from replacement human labor through machine automation.

The movement that became known as Ludism, reflecting the understandable fear of job losses in the face of technological transformations. As Shipside recounts, before the sabotage of machines by the Luddite movement led by Ned Ludd, new technology was feared and attacked, despite Marx's tentative optimism about workers distinguishing

*Corresponding author: giga.anderson@gmail.com (Anderson Santos Barroso)

Received: 16 July 2024; **Accepted:** 16 August 2024; **Published:** 29 September 2024

between machines and their exploitation by capital. However, modern Luddites still struggle to make this distinction, often expressing disgust at technology intended to make their lives easier [19]. Hundreds of years later, the discussion about the impact of technological developments still scares many professionals with the possibility of unemployment. In the scenario of multiple technological innovations, Artificial Intelligence (AI) stands out more and more [1]. AI is an area of technology focused on the development of systems and algorithms capable of performing tasks that normally require human intelligence and emerges as a contemporary technological revolution that brings with it transformative potential. Several authors approach AI through different interpretations: "The new and interesting effort to make computers think (...) machines with minds, in the full and literal sense." [8]. "Automation of activities that we associate with human thinking, activities such as decision making, problem solving, learning..." [4, 6]. "The study of how computers can do tasks that are now better performed by people." [18]. "AI… is about intelligent performance of artifacts." [13]. Among the different categories of AI, generative AIs stand out, which have revolutionized the way we conceive the creation of content and artificial interactions.

Generative Artificial Intelligences (GAIs) have the ability to generate information from existing data sets. The objective of this work is to evaluate the challenges and opportunities for technology professionals, especially systems developers, generated by the increasingly prominent presence of generative IAS as a tool for systems development. To achieve this objective, a methodological approach was adopted that consisted of a systematic mapping of the literature. The research considered recent and relevant studies that discussed the impact of GAIs on systems development and the role of software developers in this context. From now on we will talk about the related work that contributed to the construction of this study, the execution plan for the systematic literature mapping, which defines the structure and execution of our research, then we will have the results analysis section where we will do an discussion based on the content of the articles selected in the previous section. Lastly, we will highlight the threats to the validity of this work and the conclusion that portrays where this work has led us.

2. Related Works

During the conduct of this research, an article entitled Practices and Challenges of using GitHub Copilot: An Empirical Study. In this article, Beiqi et al. addresses, through an empirical study, the use of GitHub Copilot AI in integration with software development IDEs for source code autocomplete [3].

Although this work also presents a study on GAIs in the context of software development, it gives strong emphasis to the AI GitHub Copilot, not worrying about analyzing the general state of the art of GAIs.

Our work approach encompasses a broader spectrum of Generative Artificial Intelligence tools. While they explore the specific practices and challenges associated with Copilot, our work aims to evaluate opportunities and challenges offered by various generative AIs in a broader context.

Thus, while both studies share a central interest in IAG tools in software development, our work expands this perspective by offering a more holistic view of the impact of generative AIs on the work of systems developers.

3. Plan and Execution of the Systematic Mapping of Literature

A Systematic Literature Mapping (SLM) was conducted with the aim of cataloging relevant primary studies on Generative Artificial Intelligence and its challenges and opportunities for system developers. Systematic mapping is a means of evaluating and interpreting all available research relevant to a specific research question, thematic area or phenomenon of interest [10, 11]. The objective of this systematic literature mapping was formalized using the GQM model originally proposed by Basili [22]:

1. Analyze the state of the art of GAIs;
2. In order to evaluate the impact of the use of GAIs in systems development;
3. Regarding the contribution of these tools to systems developers;
4. From the point of view of systems developers;
5. In the context of professionals in the area of information technology.

The mapping process took place from August to October 2023. In this section, we will detail the MSL phases used in carrying out the research. In the first phase of the systematic mapping, two research questions (RQs) were proposed:

RQ1: Generative AIs are being applied in the systems development environment.

RQ2: AI Trends and Innovations Impacting System Developers' Opportunities and Challenges.

3.1. Literature Search

The search strategy included research conducted in online sources considering the latest academic contributions, with emphasis on updated content. During the search, 109 articles were found. We restricted the search to the last three years, but we found that most of the relevant works were published in the last year. The searches were conducted in three different digital library repositories, IEEE Xplore, Google Scholar and ACM. The choice of these repositories aimed to ensure a comprehensive compilation of information from reliable and recognized sources in the field of study that could provide a solid basis for the analysis. To develop the search strategy, we defined some initial keywords as well as synonyms. We chose to conduct the search string search in English, a decision that was strategically made based on consideration of the breadth

and depth of the scientific literature available in this language. English is widely recognized as the lingua franca of academic research, and many relevant studies, scientific advances, and developments are published primarily or exclusively in English.

Consequently, the definitive formulation of the search string was as follows:

(generative artificial intelligence OR generative AI OR applications of generative AI) AND (system development with generative AI OR challenges in using generative AI OR opportunities for system developers with generative AI) AND (recent trends in generative AI OR innovations in generative AI OR impact of generative AI on system developers OR recent advances in generative AI) AND (generative AI and system development opportunities).

3.2. Conducting the SML

A Systematic Literature Mapping (SML) was conducted with the aim of cataloging relevant primary studies on Generative Artificial Intelligence and its challenges and opportunities for system developers. Systematic mapping is a means of evaluating and interpreting all available research relevant to a specific research question, thematic area or phenomenon of interest [10]. The objective of this systematic literature mapping was formalized using the GQM model originally proposed by Basili [22]:

Publications by year

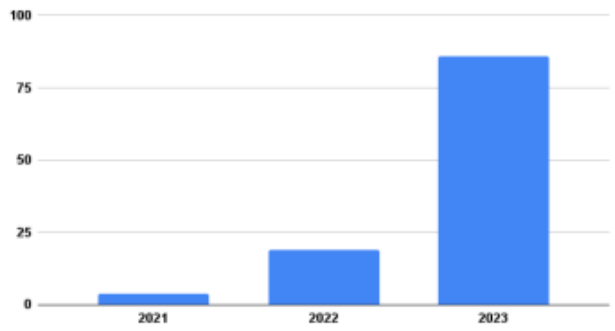

Figure 1. *Number of publications per year.*

3.3. Inclusion and Exclusion Criteria

Inclusion criteria play a fundamental role in the evaluation of each study retrieved from the search sources. Inclusion criteria (I) are used to include studies considered relevant in our systematic mapping, as follows:

I1 - the study discusses the impacts of Generative Artificial Intelligence in the systems development environment;

I2 - the study provides examples and practical cases on how generative AI is being effectively applied in the systems development environment;

I3 - the study explicitly discusses how these innovations influence the opportunities and challenges faced by systems developers.

Exclusion criteria (E) were also used to eliminate studies that do not offer significant contributions in the search for answers to the research questions:

E1 - the study is not directly related to the use of generative AI in the systems development environment;

E2 - the study does not present practical examples or concrete applications of generative IAS in the context of systems development;

E3 - the primary study is written in a language other than English;

E4 - the study does not explicitly discuss how innovations influence opportunities and challenges for systems developers.

After applying the inclusion and exclusion criteria together with an analysis of the abstracts, a total of 14 studies were selected that met the four parameters established for mapping.

Table 1. *Shows the steps used in the final selection of articles.*

Steps	Repository			Total
	AC	GS	IE	
After applying search string	38	24	47	109
After reading abstracts and introductions	5	12	31	48
After inclusion and exclusion criteria	3	6	5	14

*AC=ACM; GS=Google Scholar; IE=IEEE

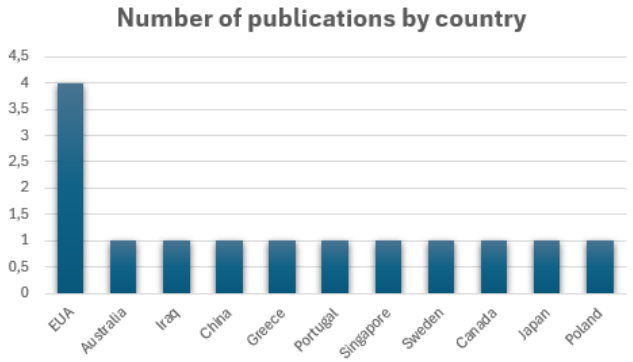

Figure 2. *Shows the distribution of the number of publications by country. There is a notable diversity of articles from several different countries, highlighting the global nature of research in this field. The United States is clearly the leader in terms of the number of publications.*

3.4. Information Extraction Strategy

For each included study, we conducted a comprehensive

assessment of the quality of the primary studies. During this process, we extracted crucial data, including the date of the study, description of the treatment of identified risks, details of the experimental design, threats to internal and external validity, study results, lessons learned, future perspectives, and any additional relevant comments. After rigorous analysis, the final selection was determined, resulting in a consolidated list of fourteen articles highlighted in Table 2.

Table 2. *Primary studies included.*

ID	Author	Year	Venue
1	Qianou Ma, Tongshuang Wu, Kenneth Koedinger	2023	Cornell University
2	Sukhpal Singh Gill, Rupinder Kaur	2023	ScienceDirect
3	Aljanabi, Mohammad &ChatGPT	2023	Mesopotamian Journal of Cybersecurity
4	A. Mastropaolo et al.	2023	45th International Conference On Software Engineering (ICSE)
5	Nascimento, Nathalia & Alencar, Paulo & Cowan, Donald	2023	Cornell University
6	Rahmaniar, Wahyu	2023	IEEE Computer Society
7	Gottlander, Johan Khademi, Theodor	2023	Chalmers Open Digital Repository
8	Wach, Krzysztof & Ejdys, Joanna & Kazlauskaite, Ruta & Korzynski, Pawel & Mazurek, Grzegorz & Paliszkiewicz, Joanna & Ziemba, Ewa & Duong, Doanh	2023	Entrepreneurial Business and Economics Review
9	Yang Ye, Hengxu You & Jing Du	2023	IEEE Computer Society
10	Beiqi, Z., Peng, L., Xiyu, Z., Aakash, A., Muhammad, W.	2023	Cornell University
11	D. Vaz, D. R. Matos, M. L. Pardal and M. Correia	2023	International Conference on Dependable Systems and Networks
12	A. Bahrini et al.	2023	Systems and Information Engineering Design Symposium
13	C. Ebert and P. Louridas	2023	ACM Transactions on Software Engineering and Methodology
14	Peter Robe, Sandeep K. Kuttal, Jake AuBuchon, and Jacob Hart	2022	European Software Engineering Conference and Symposium on the Foundations of Software Engineering

4. Analysis of Results

In this study, we conduct an analysis of the data and findings from our research on Generative Artificial Intelligence (GAI), with a special focus on the ChatGPT and GitHub Copilot models. We will explore how GAI is currently being applied in the software development landscape, highlighting significant use cases and implementations. In addition, we will discuss the latest trends and innovations in this evolving field, considering their impact on the opportunities and challenges faced by software developers. This systematic literature review provides a deeper understanding of the impact of GAI on the software development industry, as well as its implications for software developers. By examining how these technologies are shaping the software development landscape and the challenges that may arise, we aim to provide meaningful insights for those seeking to fully exploit the potential of GAI. Table 3 lists the GAIs found in this work. The diversity of countries that are discussing these tools is notable, as well as their impact on the systems development environment, demonstrating the concern for the future of this area.

Table 3. *Methods and Tools found.*

Ferramenta	ID	Country
ChatGPT	1	EUA
	2	China
	3	Iraq
	5	Canada

Ferramenta	ID	Country
	6	Japan
	7	Sweden
	8	Poland
	9	EUA
	11	Portugal
	12	EUA
	13	Greece
GitHub Copilot	1	EUA
	4	Australia
	7	Sweden
	10	EUA
	11	Portugal
	13	Greece
BERT	13	Greece
	14	Singapore

4.1. RQ1: Generative AIs Are Being Applied in the Systems Development Environment

In response to RQ1, the articles analyzed provide a comprehensive exploration of the numerous ways in which GAIs can be used in the dynamic environment of systems development. These investigations highlight how this new paradigm impacts not only the quality of deliveries, but also the productivity of professionals in the field. A crucial point emphasized is the delicate balance required between the use of these innovative tools and human action, considering the execution of tasks at different levels of complexity.

GAIs, when inserted into this scenario, present a significant potential to increase productivity in systems development. The ability to optimize routine tasks, contribute creatively and simplify processes becomes a substantial attraction for professionals in the field, who see these advances as an opportunity to improve their practices. They are also capable of generating code, generating test cases from requirements, reestablishing traceability, explaining code, refactoring legacy code, maintaining software with increased guidance, and improving existing code [5].

Generative AI, in particular, is emerging as a driving force for improving productivity in several crucial aspects of Software Engineering. One such aspect is increasing creativity, where AGI emerges as a valuable ally for developers in generating new ideas and solutions, especially in systems design and user experience (UX) development. For example, it can be employed in the creation of new designs, logos, and user interfaces, providing an innovative and efficient approach to creative challenges [5, 14].

This ability of AGI to boost creativity and suggest innovative solutions highlights the significant transformation it can bring to the field of Software Engineering. However, it is imperative to emphasize that, even with these promising advances, the balance between human intervention and automation provided by AIs is crucial to ensure quality, ethics and effectiveness in systems development practices.

In this sense, studies suggest that AIs are presented not only as facilitating tools, but as agents driving a substantial change in the way professionals in the field conceive, create and implement software solutions. The challenge lies not only in the adoption of these innovative technologies, but in the efficient harmonization between human creativity and automated effectiveness.

Another point addressed is the ethical and social issues that may arise regarding the use of these tools, including responsibility and transparency in the use of artificial intelligence in the programming scenario. In addition, the interaction between humans and these automated tools raises questions about the authorship of the generated code and the responsibility for any failures or adverse consequences.

During the analysis of the articles, the tools ChatGPT and GitHub Copilot were prominently cited by the selected works. This recurrence brought these two tools as protagonists in the issue of GAIs aimed at use in the context.

4.1.1. ChatGPT

The motivation for developing ChatGPT was to establish a powerful and flexible AI language model that could assist in a wide range of activities, such as text production, translation, and data analysis [7]. In embracing these advances, it is crucial to maintain a feedback-rich environment for AI models while ensuring that ethical considerations are adhered to. This ethical approach will determine the trajectory of code-cognition convergence in future systems development [15, 17].

Tools like ChatGPT have the potential to reshape the software development landscape. In addition to its linguistic capabilities, ChatGPT stands at the crossroads of innovation and practicality, offering solutions that can significantly enhance a developer's toolkit [17].

ChatGPT has the ability to understand and produce a wide range of words, due to its extensive training with large volumes of textual data. This linguistic transformation and its ability to generate text summaries and conduct conversations are just some of the applications that can benefit from its ability to create natural-sounding content. It is versatile enough to be trained on a variety of tasks, including language recognition, question answering, and paragraph completion. Additionally, ChatGPT plays an essential role in building chatbots and other conversational interfaces [7].

From suggesting relevant code snippets based on the current task to assisting in debugging, ChatGPT can act as a virtual programming partner. Several IDEs have explored the integration of NLP-based models2 to facilitate a more interactive and efficient approach to the coding experience.

However, relying on AI suggestions can sometimes result in less than ideal code subpatterns. This can be offset by integrating human feedback into the IDE to continually refine and train the model based on developer corrections [17].

For new developers or those transitioning to a different technology, ChatGPT can offer personalized training sessions. By simulating real-world coding scenarios and providing instant feedback, these models can facilitate hands-on learning experiences, reducing onboarding time for development teams [17].

GPT-4 can generate code from docstrings and solve coding questions in software engineering interviews at or above human performance. It can code for the front end and interact with LaTeX. It can reverse engineer code, execute Python code, and execute pseudocode. OpenAI, the company behind GPT-4, offers programmatic access to its Large Language Models (LLMs). This means that developers can not only use them in a conversational way, but also embed them in their applications. It is also possible to develop plugins, which are ways to connect the underlying models with third-party services that can answer questions and act on them [5].

This context demonstrates a practical application of ChatGPT, in its paid version, as a powerful tool for development and automation of code-related tasks. The convergence between AI and systems development is clearly marking the future, transcending traditional boundaries. ChatGPT is a notable example of this fusion, enabling industries to create intelligent and innovative solutions that combine code and cognition. However, as we move in this direction, the importance of maintaining strong ethics and an environment of constructive feedback becomes increasingly evident. This not only ensures the effectiveness of these technologies, but also ensures that they are used in a responsible and ethical manner [17, 20].

4.1.2. Github Copilot

Copilot is capable of automatically synthesizing entire functions from their signature and natural language descriptions [2].

The ability to intelligently translate natural language instructions into functional code not only saves time, but also spurs innovation by providing fast and accurate solutions. This new generation of code recommendation systems has the potential to revolutionize the way developers write code, opening the door to unprecedented productivity [2].

GitHub Copilot is available as an extension for development tools and editors such as Visual Studio Code, Visual Studio IDEs, Neovim, and JetBrains. It provides code completion powered by OpenAI Codex, a Generative AI System also powered by OpenAI [5].

In the new era of AI-enabled programming, developers must learn to properly describe the code components they are looking for in order to maximize the effectiveness of AI support [2].

It is a paradigm shift that requires a deeper understanding of the interaction between developers and AI tools. Copilot makes suggestions for code snippets, entire functions, and test cases in real time while the user is writing code. The user can also ask Copilot to generate code suggestions by writing a natural language comment within the source code [9]. This auto-completion feature of Copilot demonstrates the potential to positively increase programmer efficiency, especially in repetitive coding tasks [9].

Instead of focusing on rote, repetitive tasks, developers can direct their creativity and intellect toward more complex problems, enabling a more holistic and creative approach to systems development. Copilot represents a significant evolution in AI-assisted programming, with implications that go far beyond mere task automation. It challenges developers to communicate more effectively with AI, and in doing so, opens the door to more productive and creative programming. Code-AI collaboration is shaping the future of systems development in innovative ways, with the potential to increase the efficiency and quality of developed solutions.

4.1.3. Additional Tools

Another tool cited in the researched articles was the Generative AI BERT developed by Google in 2018. However, it is important to note that the number of citations and mentions related to BERT during this research was less relevant compared to OpenAI's ChatGPT and GitHub Copilot. This observation does not suggest a lesser importance or lesser capacity of the tool, but rather demonstrates a trend where, in the selected works, ChatGPT and Copilot gained greater visibility in the context of systems development.

4.2. RQ2: AI Trends and Innovations Impacting System Developers' Opportunities and Challenges

In the current era of AI-enabled programming, developers are faced with the need to adapt to an ever-evolving work dynamic. One essential skill they are learning is to adequately describe the code components they are seeking in order to maximize the effectiveness of AI support [2].

This is not only a technological transition, but also a transformation in the way developers interact with AI tools, involving a new set of skills that combine human knowledge with the AI’s ability to understand and generate code. Generative AI has the potential to significantly improve system productivity by automating tasks, enhancing creativity, improving accuracy and efficiency, and streamlining development processes [5].

This makes it easier to enter the Software Engineering profession by integrating these tools. While this has its advantages and disadvantages, the important thing for future engineers will be to adapt to these changes in order to benefit from the advantages. Consequently, it can reshape the role of a software engineer [9].

Nascimento et al. highlight the importance of understanding the unique capabilities of human and automated approaches. This points to the need to adjust the level of AI automation

based on the developer's experience and the quality requirements of the task at hand. It is a delicate balance between automation and human intervention, aiming at optimizing productivity and quality in software engineering [12].

Ebert and Louridas point out that answers and solutions are generated by probability-based models, not necessarily found in some reliable source. This means that they can be wrong. AI tools can hallucinate, responding in a wildly erroneous way, while at the same time being extremely confident that they are right. Things are improving (GPT-4 seems to be better than its predecessors), but the user should always verify the answers. Relying on AI tools for tasks where you cannot determine the correct answer or how to verify it can lead to complications and pitfalls. For software development, this means that human supervision and intervention, such as reviews, are required [5].

Although there is speculation that AI-based computing could increase productivity and eventually replace software engineers in systems development, there is currently a lack of empirical evidence to support this claim [12].

Most studies are still ongoing, and the application of machine learning techniques in software engineering needs to be carefully examined, especially with respect to different non-functional properties. Tests are needed to evaluate Language Modeling Language-Assisted Programming. Further comparisons can be drawn between human-human vs. human-AI pair programs in programming. Furthermore, further work can explore how best to support LLM-Assisted Programming with insights from the rich literature on the nature of human-human pair programming [16].

ChatGPT, in turn, presents both opportunities and challenges for system developers, opening up prospects for AI-assisted coding, debugging, and more [17].

This, however, raises ethical dilemmas, economic implications, and critical security considerations that require a careful approach in implementation. The study by Nascimento et al. reveals that automated systems such as ChatGPT can outperform novice software engineers on specific tasks. This superiority is particularly notable on easy- and medium-level problems, demonstrating the effectiveness of AI in these contexts. However, AI performance still does not outperform experienced programmers in competitions in terms of solution performance, highlighting the need for balanced collaboration between humans and AI [12].

According to authors Gottlander and Khademi, there is still some hesitation and concern surrounding the integration of AGI tools in industry, and as a result, regulations and guidelines on how to use and develop AGI tools will need to be clearly implemented before they can be integrated and fully trusted [9]. As developers integrate generative AI into their workflows, the need arises to find a harmonious balance between the efficiency of this model and maintaining the distinct human touch that is essential to systems development [17].

The challenge is to leverage the advantages of AI while maintaining the human creativity and decision-making that are inherent in many aspects of software engineering. Addressing the ethical and legal issues, Wach et al. argue that advocating for regulation of the AI market is crucial to ensure a level playing field, promote fair competition, and protect intellectual property and privacy rights, and prevent potential geopolitical risks. To mitigate risks related to lack of information quality control, misinformation, deepfake, and algorithmic bias, the use of diverse and high-quality, pre-approved datasets and the implementation of human feedback loops are recommended. Software engineering for and with AI should begin with statements that create boundaries of what is allowed and what is not [5].

Rahmaniar adds that as AI models play a more significant role in systems development, the ethics of AI-generated code should be closely monitored. Future iterations of ChatGPT will likely need to incorporate ethical guidelines, ensuring that the generated code complies with privacy, security, and fairness standards. This ethical perspective is essential to guide the continued advancement of AI in software development, respecting fundamental values and promoting social welfare [17].

In a rapidly changing job market, where AI training becomes a prominent job category, professionals in the field must continue to acquire new digital skills through education and retraining. This adaptation to new opportunities is essential to remain relevant and effective in the new era of AI-enabled programming [21].

It is a crucial step to embrace change and thrive in an ever-evolving technology landscape.

In addition to exploring promising advances in the use of code analysis and generation tools, it is imperative that practitioners be aware of the critical privacy and security implications of this process. Practitioners should also be aware of the privacy and security implications. When using a tool to analyze your code, you should be careful about what happens to that code. If your code is open source, it probably doesn't matter that it may leave traces in the tool's models, or even be used as training material for the tool itself. But with proprietary code, you may not want your code to leave the confines of your private repositories. Different tools offer different safeguards for this; you should read their terms of use carefully. It is difficult or impossible to tell what is original work and what is generated fake, such as photos and videos used for disinformation. Requiring a source declaration will not work, because those who wish harm do not follow these self-imposed rules [5].

This scenario reinforces the importance of balanced collaboration between humans and AI in software engineering, guided by ethical principles, in a constantly evolving technological landscape. A cautious approach is necessary to face the challenges, maximize the opportunities and ensure a harmonious integration of AI in systems development.

5. Threats to Validity

This section emphasizes the critical importance of a

thoughtful approach to addressing the considerations that must be thoroughly analyzed and confronted in future replications of this study. The underlying intent is to enhance not only the robustness but also the broader applicability of the results obtained. The organization of the threats to validity into two distinct categories, Internal Validity and External Validity, is intended to provide structured clarity and establish a solid basis for the assessment and mitigation of potential sources of uncertainty.

The attention given to these concerns is intended to ensure that the methodology employed is not only rigorous but also capable of withstanding future replication, ensuring the reliability and consistency of the results. By classifying the threats to validity into specific categories, we seek not only to identify areas of potential vulnerability but also to outline proactive strategies to address these challenges, thereby establishing a robust foundation for subsequent research.

Internal Validity: During the selection and data extraction, some articles may not have provided detailed information about their objectives and results. This may make it difficult to apply the inclusion and exclusion criteria consistently. To address this issue, the selection and data extraction were conducted in a manner that ensured that any disagreements or conflicts were carefully discussed and resolved among those involved in the study. This collaborative approach ensured a comprehensive and objective assessment of the articles while maintaining the integrity of the internal validity of the study.

External Validity: The search process outlined in Section 2 was developed after extensive literature mapping and was validated with the unanimous consensus of all those involved in the study. We tested the comprehensiveness and representativeness of the retrieved studies, both manually and automatically, to ensure that our search covered a diverse and representative sample of the available literature. While some articles may not be available due to access or cost constraints, we have taken steps to ensure that our sampling is as inclusive as possible [21].

This study acknowledges and actively addresses threats to validity that may arise in future replications. Our commitment to methodological rigor and collaborative approach to data mapping and selection aims to ensure the quality and reliability of our findings, allowing them to be broadly applicable and reliable in the research community.

6. Conclusions and Future Work

This study comprehensively explored, through a systematic mapping of the literature, discussions on the impact of Generative Artificial Intelligence on the systems development environment. The analysis addressed significant opportunities and challenges for professionals in the field. The results obtained highlight the relevance that these technologies have been gaining in the productivity and optimization of day-to-day solutions, as well as in driving innovations. The studies also emphasize the need for a balance between the use of AI tools and development carried out by human participation. The ethical and legal issues involving the use of Generative Artificial Intelligence by systems developers also require attention. However, there is still a lack of empirical evidence regarding the replacement of human labor by automation provided by machine work. The topic still needs to be rigorously addressed and discussed in the future and requires further studies and in-depth discussions before confirming or discarding this theory. This study not only contributes to the current landscape of debates on AIs, but also highlights the complexity of the landscape, emphasizing the importance of a cautious and reasoned approach to the implementation and continuous evolution of these technologies in the field of systems development. By offering a comprehensive and detailed perspective, our aim is to provide support to both established practitioners and researchers seeking to continuously explore the vast and dynamic field of Generative Artificial Intelligence. This study not only contributes to the current understanding, but also encourages continued reflection and a reasoned approach to address the challenges and seize the opportunities that arise in this constantly evolving landscape.

Abbreviations

AI	Artificial Intelligence
GAI	Generative Artificial Intelligence
SML	Systematic Mapping of Literature
UX	User Experience
DEEPFAKE	Technique to Alter a Video or Photo with AI

Acknowledgments

We thank the Computer Science Department of Tiradentes University for supporting this work.

Author Contributions

Samira Santos Caduda: Data curation, Formal analysis, Investigation, Validation, Visualization, Writing – original draft

Anderson Santos Barroso: Conceptualization, Methodology, Supervisão, Writing – review & editing

Funding

This work is not supported by any external funding.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] A. Bahrini et al., "ChatGPT: Applications, Opportunities, and Threats", 2023 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2023, pp. 274–279, https://doi.org/10.1109/SIEDS58326.2023.10137850

[2] A. Mastropaolo et al., "On the Robustness of Code Generation Techniques: An Empirical Study on GitHub Copilot", 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), Melbourne, Australia, 2023, pp. 2149-2160, https://doi.org/10.1109/ICSE48619.2023.00181

[3] Beiqi, Z., Peng, L., Xiyu, Z., Aakash, A., Muhammad, W. (2023). Practices and challenges of using Github copilot: An empirical study, arXiv: 2303.08733, https://doi.org/10.48550/arxiv.2303.08733

[4] Bellman, R. E. (1978). An Introduction to Artificial Intelligence: Can Computers Think? Boyd & Fraser Publishing Company. ISBN 9-78087-835-0667.

[5] C. Ebert and P. Louridas, "Generative AI for Software Practitioners," in IEEE Software, vol. 40, no. 4, pp. 199-200. 30-38, July-August. 2023, https://doi.org/10.1109/MS.2023.3265877

[6] D. Vaz, D. R. Matos, M. L. Pardal, and M. Correia, "Automatic Generation of Distributed Algorithms with Generative AI,"2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplementary Volume (DSN-S), Porto, Portugal, 2023, pp. 127-131, https://doi.org/10.1109/DSN-S58398.2023.00037

[7] Gill, Sukhpal Singh & Kaur, Rupinder. (2023). ChatGPT: Vision and Challenges. https://doi.org/10.1016/j.iotcps.2023.05.004

[8] Haugeland, J. (Ed.). (1985). Artificial Intelligence: The Idea Itself. MIT Press. ISBN 0-262-58095-0.

[9] J. Gottlander and T. Khademi, "The Effects of AI-Assisted Programming on Software Engineering: A GitHub Copilot's Observation in Industry," in Neurocomputing—Algorithms, Architectures, and Applications, F. Fogelman-Soulie and J. Herault, eds., NATO ASI Series F68, Berlin: Springer-Verlag, pp. 227–236, 2023. https://doi.org/ 20.500.12380/306738

[10] Kitchenham, Barbara. (2004). Procedures for Conducting Systematic Reviews. Keele, UK, Keele Univ. ISSN: 1353-7776.

[11] M. Aljanabi, "ChatGPT: Future Directions and Open Possibilities", in Mesopotamian Journal of Cybersecurity, vol. 1, no. 1, 2023. (Editorial article style). https://doi.org/10.58496/MJCS/2023/003

[12] Nascimento, Nathalia & Alencar, Paulo & Cowan, Donald. (2023). Comparing Software Developers with ChatGPT: An Empirical Investigation. arXiv preprint arXiv: 2305.11837 (2023). https://doi.org/10.48550/arXiv.2305.11837

[13] Nilsson, N. J. (1998). Artificial Intelligence: A New Synthesis. Morgan Kaufmann. ISBN 1-55860-535-5.

[14] Norvig, Peter. Artificial Intelligence. Available at: My Library, (3rd edition). GEN Group, 2013. ISBN 978-1558605350.

[15] Peter Robe, Sandeep K. Kuttal, Jake AuBuchon, and Jacob Hart. 2022. Pair Programming Conversations with Agents vs. Developers: Challenges and Opportunities for the SE Community. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022). Association for Computing Machinery, New York, NY, USA, 319–331. https://doi.org/10.1145/3540250.3549127

[16] Qianou, & Ma, & Wu, Tongshuang & Koedinger, Kenneth. (2023). Is AI the Best Programming Partner? Human-Human Pair Programming vs. Human-AI Pair Programming. https://doi.org/10.48550/arXiv.2306.0515

[17] Rahmaniar, Wahyu. (2023). ChatGPT for Software Development: Opportunities and Challenges. https://doi.org/10.36227/techrxiv.23993583.v1

[18] Rich, E. and Knight, K. (1991). Artificial Intelligence (second edition). McGraw-Hill. https://doi.org/10.4236/ojapps.2019.910062

[19] Beside the Ship, Steve. Karl Marx's Capital: A Modern and Practical Interpretation (Classics of Economic Thought). Available at: Minha Biblioteca, Editora Saraiva, 2010. ISBN 13. 978-8502102606.

[20] Yang Ye, Hengxu You & Jing Du. 2023. Improving trust in human-robot collaboration with chatgpt. https://doi.org/10.1109/ACCESS.2023.3282111

[21] Wach, Krzysztof & Ejdys, Joanna & Kazlauskaite, Ruta & Korzynski, Pawel & Mazurek, Grzegorz & Paliszkiewicz, Joanna & Ziemba, Ewa & Duong, Doanh. (2023). The dark side of generative artificial intelligence: A critical analysis of controversies and risks of ChatGPT. Entrepreneurial Business. https://doi.org/10.15678/EBER.2023.110201

[22] Olbrich, S., Cruzes, D. S., Basili, V., and Zazworka, N. (2009). The Evolution and Impact of Code Smells: A Case Study of Two Open Source Systems. InProc.of the 2009 3rd Int. Symposium on Empirical SoftwareEngineering and Measurement, pages 390–400. IEEEComputer Society. http://doi.org/10.1109/ESEM.2009.5314231

Research Field

Samira Santos Caduda: Generative Artificial Intelligence, Artificial Intelligence, Machine Learning, Software Engineering

Anderson Santos Barroso: Human factors in computing, Software Engineering, Data Mining, Experimental Engineering