Applications of Generative Artificial Intelligence in the Software Industry

Ivo Damyanov [1], Nikolay Tsankov [2], Iliya Nedyalkov [3]

[1] *Faculty of Mathematics and Natural Sciences, South-West University, Blagoevgrad, Bulgaria*
[2] *Faculty of Education, Trakia University, Stara Zagora, Bulgaria*
[3] *IDNC Soft Ltd, Sofia, Bulgaria*

Abstract **– The increasing demands on software development are putting serious pressure on its pace. To assist software developers, an increasing number of tools powered by generative artificial intelligence are being introduced. This paper aims to investigate how the use and integration of generative AI have evolved among professionals in the software industry, based on a study involving 104 individuals working in Bulgarian software companies. Data was collected in April 2024 through an online questionnaire with four separate groups of questions related to the use of generative AI at work. The study found that 2/3 of the respondents use generative AI actively in their daily work. They highly value the practical benefits of this type of technology, which most often consist of automating routine activities, accessing information quickly, generating initial code, and writing documentation. As a result of these benefits, developers are increasingly moving towards using generative AI at the expense of professional support platforms. The main benefits they cite include faster solutions, more specific and relevant answers, and significantly shorter time to reach the desired outcome.**

Keywords **– Generative artificial intelligence, ChatGPT, Copilot, software development, software developers support forums.**

DOI: 10.18421/TEM134-10
https://doi.org/10.18421/TEM134-10

Corresponding author: Ivo Damyanov,
Faculty of Mathematics and Natural Sciences, South-West University, Blagoevgrad, Bulgaria
Email: damianov@swu.bg

Received: 04 June 2024.
Revised: 18 October 2024.
Accepted: 04 November 2024.
Published: 27 November 2024.

1. Introduction

General pretrained transformer - the de facto standard today for creating large language models - has become the fastest technology to reach over 100 million users, thanks to OpenAI's ChatGPT [1]. The development of an architecture based solely on attention mechanisms by researchers at Google [2] led to a boom in the creation of platforms and their implementation as various products and solutions available to users. Trained on large language corpora, general pretrained transformers were made accessible through API or chat-based interfaces to anyone wishing to utilize these resources to enhance their work.

Software is a fundamental factor in the modern industry and economy, and the striving for a real-time economy puts serious pressure on the pace of software development, the maintenance of the entire infrastructure (not just the software) and most importantly, the ability of teams to cope with rapidly changing tasks. With the development of cloud technologies and the migration of infrastructure solution to the cloud, the automation of basic processes and repetitive operations allows developers to be significantly more productive. However, building software is a creative process that takes time. The introduction of collaborative tools, the improvements of integrated development environments (IDE), the emergence of package managers such as npm, nuget, and others, in response to the increasing complexity of projects and the need for effective dependency management are just a small part of the technological solutions in support of accelerated software development. To support their search for solutions to the various problems they encounter, software engineers have been using for years professional Q&A forums like Stack Overflow [3], and the available online documentation for software platform and software framework. Last but not least they use a simple web search of the Internet in the millions of specialized content pages.

Software complexity falls into two main categories - essential complexity and accidental complexity [4]. Essential complexity is determined by how hard the problem itself is and arises from its characteristics and requirements. Such complexity is intrinsic to the problem and cannot be eliminated by technological or methodological means as it is directly related to the nature of the problem. Accidental complexity, on the other hand, refers to the complexity introduced by the choice of technology or approach adopted to solve a problem. This kind of complexity results more from the decisions made in the software development process than from the nature of the problems being addressed. It can be managed and reduced with appropriate techniques, tools, and practices. The increasing size and complexity of software systems make conventional method of working too error-prone. In software engineering, managing complexity is crucial for fostering well-structured and maintainable systems. This is achieved through the creation of composite objects, which act as fundamental building blocks for modular architectures. By promoting modularity, developers gain a high-level understanding of the application, facilitating efficient comprehension and long-term maintainability. Furthermore, managing software complexity has led to the creation of numerous general-purpose and domain-specific languages, platforms and software frameworks. This proliferation increases the range of development skills that developers need to acquire and requires them to quickly learn these tools to enhance or maintain existing software systems.

The drive to automatically generate programming code is not a recent trend. Many researchers have proposed various mechanisms for mixed - manual and automatic code generation based on software templates [5], on metadata [6], on domain models [7] or mediated by domain-specific languages [8].

A number of studies have demonstrated the controversial achievements of generative artificial intelligence in software engineering [9]. Generative AI produces different answers to the same queries - it is nondeterministic in nature. Therefore, researchers' efforts are primarily focused on making it produce usable and reliable answers that engineers can use directly. Although problems with hallucinations [10] continue to be a challenge, in the formal world of software development, minimizing them is expected to be easier, but still not without difficulties due to the sensitivity to input parameters and settings. Progress in dealing with hallucinations has been steady, for example researchers at Google Deep Mind, Sandford University and University of Illinois at Urbana-Campaign have developed the Search-Augmented Factuality Evaluator (SAFE) tool [11] to fact-check the results of large language models.

IDEs are continually improving to assist programmers in creating code. Advances in these environments are aimed at increasing productivity and improving the quality of the source code produced. Code auto-completion, syntactic coloring, code navigation, various refactoring, and static analysis tools are now standard features that have been available for over a decade. Integrated development environments also became some of the first platforms in which AI-based solutions were embedded. This included features such as automatic code suggestion, code error prediction, performance optimization, and more, and software engineers readily adopted them. Tools such as OpenAI's Codex, Tabnine, Github Copilot [12] bring a whole new level to the service of accelerated code writing.

Regardless of how generative AI is used (through some integrated environment or through a chat client) it is gradually proving its efficacy. It is therefore of interest to us to find out, nearly a year and a half following ChatGPT's surge in popularity, to what extent AI adoption has altered the practices in the software industry. Moreover, it is important for us to evaluate the extent to which traditional approaches remain relevant for developers or whether there is a significant shift in favor of AI-driven tools.

The natural evolutionary processes of creating tools to help developers also have their continuation in the announced collaboration between OpenAI and Stack Overflow. Their goal is to provide users and customers with the accurate and vetted data foundation that AI tools need to find a quick solution to a problem so that technologists can stay focused on priority tasks. OpenAI will also surface validated technical knowledge from Stack Overflow directly into ChatGPT, giving users easy access to trusted, attributed, accurate, and highly technical knowledge and code backed by the millions of developers that have contributed to the Stack Overflow platform for 15 years [13].

In a comparative study conducted by Xu *et al.* [14] between answers given by generative AI and humans in professional support forums, it was found that currently generated answers by ChatGPT are of lower quality compared to those generated by humans for all aspects considered in the study, namely correctness, usefulness, diversity, readability, clarity, and conciseness. A convergence in the responses of generative AI and humans in terms of diversity, readability, and clarity has been found, leading the authors to consider the potential development opportunities for both solution search capabilities.

The research interest focuses on this process of ubiquitous AI in mainstream activities and the way it is being adopted/recognized in public or industrial sectors. To this end, the following questions were set to seek answers to:

RQ 1: To what extent do professionals in the software industry, particularly software developers, incorporate generative AI into their daily tasks?
RQ 2: How do they evaluate the benefits of using generative AI?
RQ 3: What are software developers' preferences when it comes to using generative AI compared to modern developer support forums like Stack Overflow?
RQ 4: What are software developers' views on the future of their profession, and what concerns or desires do they have regarding upskilling to better utilize generative AI solutions in their work?

The attention is focused on Bulgarian software developers because Bulgaria has proven its ability to adapt in the software industry. The ICT industry in the country comprises about 10,000 companies, 70% of which are exporters, which puts the country in the top outsourcing destinations [15]. With a steady and stable growth, the ICT industry sector in Bulgaria is expected to generate more than 4 billion euros in revenue by 2023 [16]. Generative AI can contribute to the development of the ICT sector in Bulgaria by helping companies automate processes, improve efficiency, and create new products and services.

2. Related Work

Examining the main databases with scientific publications Scopus and Web of Science, as well as the publications indexed in Google Scholar, there is a significant growth in the scientific literature dedicated to the study of diverse applications of artificial intelligence in various aspects of human activity. Some authors examine the role of generative AI in the various stages of software development, focusing on applications such as automated code generation and process optimization. For example, Sauvola *at al*. [17] consider four scenarios for using generative AI in software development operations, namely: (1) traditional software development operations: where humans own all roles, and development tools and environments provide automation; (2) AI in a loop: where humans dominate AI, but AI begins to manage larger and more complex work domains; (3) AI assumes role(s): where AI starts to assume selected roles, for example, AI is used for process management, design, implementation, testing, delivery, and support; (4) Human-in-the-loop: where AI manages development operations in various roles and humans monitor and control the process.

Other authors mainly explore the benefits and concerns of using ChatGPT and Copilot in software development projects by software engineers during their training [18], [19] and the challenges faced by novice programmers. In [20] the authors thoroughly evaluate the effectiveness of ChatGPT in solving coding problems at various levels of complexity and popularity. They also explore what programmers' attitudes are about ChatGPT and other AI tools. These studies highlight the potential of artificial intelligence to transform the software development process by making it more efficient and accessible.

Although still in the early stages of adopting AI as a tool in software development, clear trends are already emerging toward using AI as an initial developer support option. It becomes an important ally in the software development process, to developers. This means that it has the potential to displace the traditional forms of support developers have received from public code repositories (such as GitHub) and professional forums (such as Stack Overflow).

The current study complements existing research by capturing the current usage of AI tools among software industry professionals in Bulgaria and comparing their views on the benefits of AI tools with traditional developer tools.

3. Materials and Methods

To evaluate the extent to which generative AI is integrated in the daily tasks of Bulgarian software developers, a 22-question survey was created using Google Forms. The questionnaire was sent to over 30 small, medium, and large software companies, with teams in various cities across the country. Participation was entirely voluntary and responses were submitted anonymously. Participants were given one month (April 2024) to complete the survey.

In designing the current survey among IT companies, the strategic approach was aimed at carefully balancing data collection needs and respecting company privacy policies. In this way, the focus was primarily on aspects of IT professionals' work with generative artificial intelligence, allowing the study to concentrate on the key competencies and motivations of the participants while ensuring compliance with ethical principles.

Only two questions are used to profile the respondents in terms of their position and years of experience in the IT industry, particularly in programming. The survey questions can be thematically grouped as follows:

- Main position and work experience;
- Degree of use of AI applications and familiarity with various implementations of generative AI;
- Specific tasks solved with AI;
- Potential benefits associated with using generative AI;
- Comparison of using generative AI and professional forums to support developers.

Some of the questions include "Other" as a possible answer, allowing respondents to provide specifics. One of the questions also allows for a short text response. Descriptive statistics and correlation analysis were used to analyze the survey data.

4. Results

The profile of respondents primarily includes software developers with diverse specializations including full-stack, back-end and front-end which accounts for approximately 57% of the respondents. Additionally, 11% are software quality assurance specialists, technical or project managers (~19%), DevOps (~7%), and 6% hold other positions.

Professionals with years of experience were more likely to participate in the survey (Figure 1), with those with more than 10 years of programming experience at a rate of 43.27%, The average level of experience is 5-10 years.

Not all survey participants reported using generative AI in their work. This includes personal projects, learning new programming languages, environments, frameworks, libraries, and packages. However, the results are positive: 73.1% of respondents indicated they do use generative AI. Those who answered positively were asked further questions to delve deeper. Conversely, respondents who said they do not use generative AI were offered the chance to share their views on the value of forums and professional support sites like Stack Overflow. This group was also asked about their feelings regarding the potential threat of generative AI to their jobs, and their openness to retraining in light of this new technology.

In terms of familiarity with some of the more popular generative AI platforms to date, the results indicate ChatGPT as the top favorite among developers (M=4.42 out of 5), followed by Copilot (M=3.22 out of 5), with Pi.AI being the least familiar (M=1.25 out of 5) (Table 1). The question was only asked to those who answered positively that they use AI in their work.

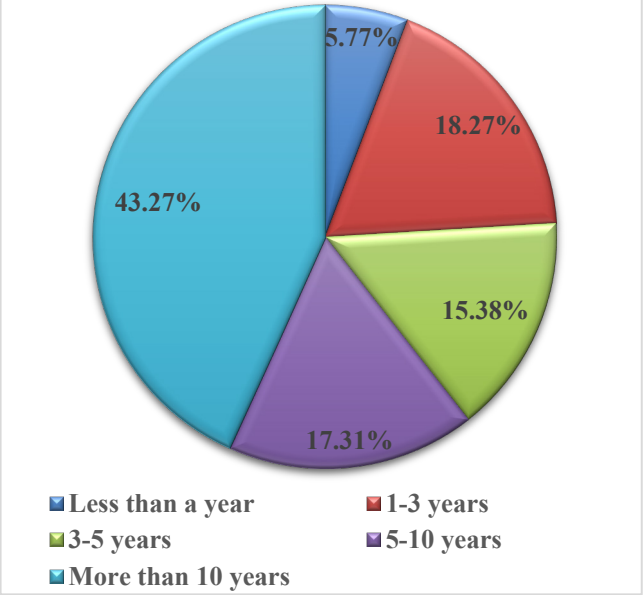

Figure 1. Participants' experience distribution

In terms of frequency of use, the mode and median indicate that IT professionals use generative AI predominantly at a frequency of "a few times a week." Spearman's correlation result showed that there was no significant correlation between years of experience and frequency of use, r(74) = 0.174, Sig .134.

Frequency of use of AI stands out in the responses, with 30.3% of respondents using it "daily" and a not insignificant 42.1% "several times a week" (Figure 2). This undoubtedly leads to a deeper understanding of the benefits of AI for their daily activities, but also to an awareness of the drawbacks associated with it. The use of generative AI can be extremely beneficial to programmers in several key ways that aid their learning in new programming concepts and practices, and this was reported by 59.3% of them.

Table 1. Familiarity with generative AI platforms

Degree of familiarity with different generative AI platforms	Unknown to me	I've heard of it but haven't used it	I've tried it	I use it sometimes	I use it often	Mean	Median	Standard deviation
	1	2	3	4	5			
BgGPT	24	38	9	5	0	1.93	2	0.84
ChatGPT	0	0	7	30	39	4.42	5	0.66
Copilot	1	29	16	12	18	3.22	3	1.23
Bard/Gemini	6	42	19	7	2	2.43	2	0.87
Perplexity	58	14	3	1	0	1.3	1	0.61
Pi.AI	60	14	1	1	0	1.25	1	0.54

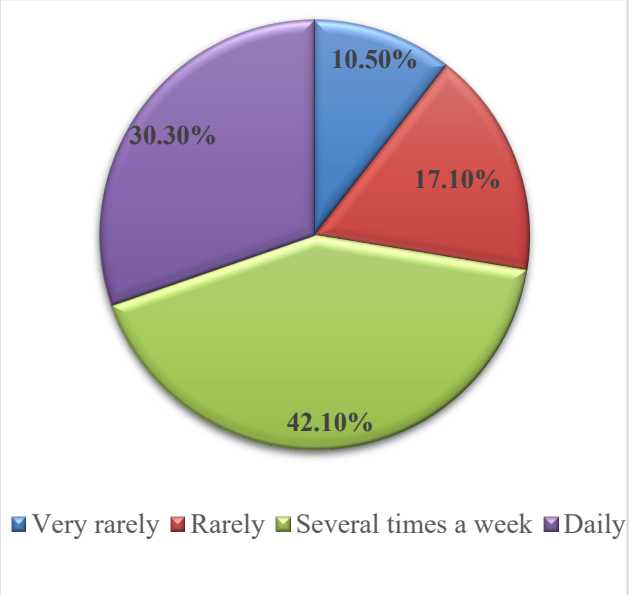

Figure 2. Frequency of generative AI use

Generative AI can provide instant answers and explanations to programmers' questions, enabling real-time assistance that is especially valuable when learning new technologies or troubleshooting code issues. Additionally, AI can generate sample codes and templates that demonstrate how to use specific concepts or libraries, helping programmers understand new approaches and techniques for applying theoretical concepts in practice. Another significant benefit of generative AI is its ability to suggest optimizations for existing code and recommend best practices for writing more efficient and readable code. This capability helps programmers improve their coding skills and avoid common mistakes.

Furthermore, generative AI can personalize learning by offering tools and tasks tailored to an individual's current level and learning goals, thereby accelerating the acquisition of new programming knowledge and skills.

It is statistically confirmed that knowledge expansion and deeper understanding are directly related to the use of generative AI when it helps to learn new concepts and practices in programming (Figure 3). Spearman's correlation coefficient is 0.580 (Sig. 0.01).

Boosting programmer efficiency and productivity can be achieved by automating routine tasks with AI. This frees up valuable time for programmers to learn new concepts and technologies. Survey results reveal that the most common uses of generative AI in daily activities (both professional and personal) are: Automating routine tasks (56.6%): Generative AI can handle repetitive tasks, allowing programmers to focus on more strategic work; Quick reference for online documentation (55.3%): Accessing vast databases of documentation, training materials, and examples eliminates the need for time-consuming searches across multiple sources. Following these top uses are: Generating initial code (47.4%): AI can provide a starting point for code, streamlining the development process; Writing documentation (44.7%): Generative AI can assist in creating clear and concise code documentation; Problem-solving guidance (39.5%): AI can offer advice on how to approach specific programming challenges. These rationales highlight how generative AI can be a powerful tool for learning and development in programming by providing individualized help, examples, and optimizations that accelerate and facilitate the learning process, knowledge expansion, and deeper understanding.

A relatively smaller proportion of people indicated that they relied on generative AI to search for problems or implement optimal code, create tests, or generate code descriptions.

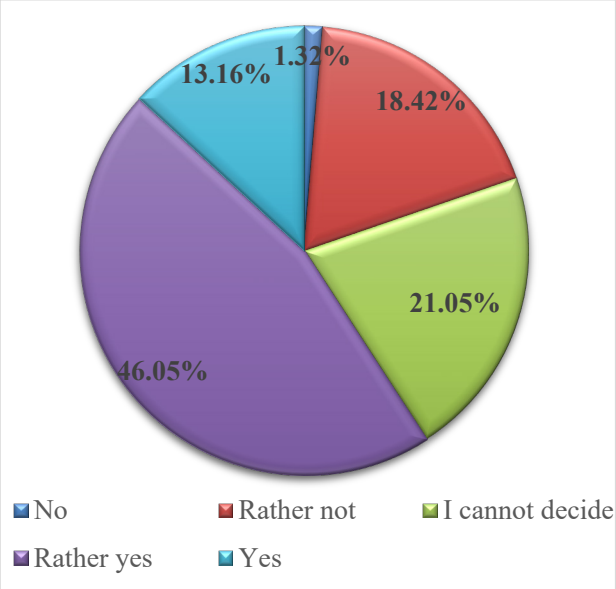

Figure 3. Perceived helpfulness of generative AI in learning programming concepts

Undoubtedly, AI can automate many routine and repetitive tasks that currently require human intervention. This could lead to a decreased demand for programmers in certain activities and positions. Some basic roles, such as maintenance and minor code changes, might be entirely replaced by AI, reducing the number of entry-level programming jobs. The rapid development of AI technologies may render some skills and knowledge obsolete, necessitating that programmers continually learn and adapt to new technologies and approaches within the context of lifelong learning. The use of AI in programming also raises several ethical and professional questions, such as responsibility for errors or issues in AI-generated code, as well as data security and confidentiality concerns, which require new protective measures and practices. The ability of AI to generate code and solutions might reduce programmers' creative involvement in the development process and foster a certain level of dependency on AI tools, potentially limiting their inclination to experiment with new ideas and programming solutions. A key factor in the future will be programmers' ability to adapt to the changing environment, acquire new skills, and find ways to use AI as a tool that complements, expands, and enriches their capabilities rather than replacing them.

Of particular interest to us is how the perception of benefit derived from generative AI changes depending on what experience professionals have. One question presents the benefits in three groups:

- **Cognitive** - expanding knowledge, deepening understanding, stimulating critical thinking;
- **Practical** - increased efficiency, increased effectiveness, improved communication, increased creativity
- **Emotional** - increased motivation, reduced stress.

The results are as follows: Cognitive - 55.6%, Practical - 88.9%, Emotional - 16.7%. The total exceeds 100% because respondents were allowed to select multiple answers (Table 2).

In order to test the statistical significance between the different stated benefits using a non-parametric Chi-Square test on the data collected from the respondents (Table 3), it is found that the highest statistically significant relationship is between cognitive and practical, then between cognitive and emotional and the lowest between practical and emotional.

Table 2. Descriptive statistics on benefit derived from generative AI

	Frequency	%	% of Cases
Practical	64	55.17%	84.21%
Cognitive	40	34.48%	52.63%
Emotional	12	10.34%	15.79%
Total	116	100%	

Table 3. Statistical significance of benefits derived from generative AI using Chi-Square test

	Chi-Square	df	p
Cognitive - Practical	12.07	1	.001
Cognitive - Emotional	7.65	1	.006
Practical - Emotional	5.2	1	.023

Since each option was chosen by a different number of respondents, the data was normalized by presenting it as percentages scaled according to the total number of respondents. From the obtained chart (Figure 4), several observations can be made: the lines for cognitive and emotional benefits (plotted against years of experience) run parallel, while the line for practical benefits exhibits the opposite behaviour—when cognitive and emotional benefits increase, practical benefits decrease, and vice versa. For specialists with the most experience, the three lines almost converge.

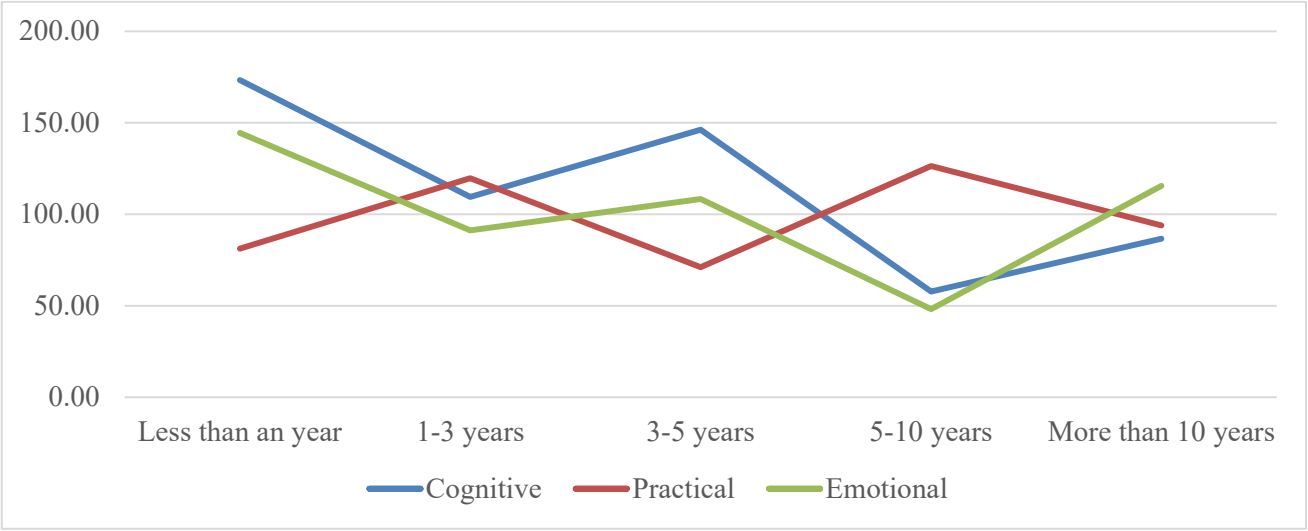

Figure 4. Normalized representation in percentages of perceived benefits from generative AI by experience level, scaled to total respondents

Generative AI enters into direct competition with developers support forums necessitates evaluating its impact on IT specialists' workflows. The study compared these tools on some key aspects: frequency of use, information search efficiency, solution speed, accuracy, user satisfaction, and work scalability.

When comparing frequency of use, the evaluation of time saved, and use of AI versus professional support forums (Table 4), a statistically significant moderate correlation is observed between the frequency of AI use and its use compared to professional forums.

Table 4. Spearman's rank correlation of generative AI impact on IT workflows

			Frequency of use	Estimation of time saved	Using AI vs. Forums
Spearman's rho	**Frequency of use**	Correlation Coefficient	1.000	.312**	-.406**
		Sig. (2-tailed)	.	.006	.000
		N	76	76	76
	Estimation of time saved	Correlation Coefficient	.312**	1.000	-.310**
		Sig. (2-tailed)	.006	.	.006
		N	76	76	76
	Using AI vs. Forums	Correlation Coefficient	-.406**	-.310**	1.000
		Sig. (2-tailed)	.000	.006	.
		N	76	76	76

**. Correlation is significant at the 0.01 level (2-tailed).

The following conclusions can be drawn from the analysis (Table 5): respondents use AI tools and online forums almost equally (M=2.21 out of 3, SD = 0.79), with a slight preference for online forums; regarding effectiveness, there is no difference and respondents rate them as equally effective (M = 2 out of 3, SD = 0.89); the score for finding a solution quickly is in favor of the generative AI (M= 1.78 out of 3, SD = 0.87), and also the accuracy of the solutions proposed by the AI is better than what they manage to find in the forums (M= 2.36 of 3, SD=0.81).

This also shaped greater satisfaction with AI, with the perception of better job scaling again favoring generative AI (M = 1.58 out of 3, SD = 0.75).

The results obtained in terms of time saved by using generative AI in the work of software developers reveal a significant potential for the development of the capabilities and application of AI, given the moderate variation in responses and the highest percentage rating - moderate time savings (from 10% to 30%).

The last two questions in the survey are about how software developers see their profession in the context of the pervasiveness of AI in the software industry. Only 22% either cannot judge or feel any threat. Also, only 25% cannot assess or have a negative attitude towards possible retraining.

Table 5. Comparison between professional Q&A forums and generative AI

	N	Mean*	Median*	Standard deviation
How often do you use generative AI tools compared to professional online forums (e.g. Stack overflow)?	76	2.21	2	0.79
How do you rate the effectiveness of generative AI tools compared to professional online forums for searching for information?	76	2	2	0.89
How do you rate how quickly you find a solution to a problem using generative AI tools compared to searching professional online forums?	76	1.78	1	0.87
How do you rate the accuracy of proposed solutions from generative AI tools compared to those found in professional online forums?	76	2.36	3	0.81
How do you rate your satisfaction with using generative AI tools compared to professional online forums?	76	1.76	2	0.83
How do you rate how easily your work can scale with generative AI tools compared to professional online forums?	76	1.58	1	0.75

5. Discussion

The results of the survey show that generative AI is rapidly entering the work of software professionals, with a significant percentage of them (73.1%) already using such tools in their professional tasks. This is particularly noteworthy given the profile of the respondents, who span a wide range of IT roles — predominantly software developers, as well as quality assurance specialists, technical managers, and DevOps engineers. A large percentage of respondents have solid experience (43.3% with more than 10 years in the field), making the adoption and use of AI among these professionals even more interesting and significant.

One of the most intriguing aspects of the research is the high adoption rate of ChatGPT (M = 4.42 out of 5), which demonstrates its dominant role as a generative tool among developers. Although other platforms, such as Copilot, also find a place in developers' work (M = 3.22 out of 5), lesser-known tools such as Pi.AI and Perplexity remain underutilized. This reveals the challenge for new AI platforms to establish themselves in an already competitive market.

The primary motivation for using generative AI is to improve efficiency and productivity. The results indicate that a large percentage of programmers (56.6%) use AI to automate routine tasks, allowing them to focus on more complex and creative work. AI is also used as a tool for accessing documentation (55.3%) and generating sample code (47.4%), proving crucial in speeding up workflows. These findings support the conclusion that AI can increase productivity by saving time and effort for programmers in performing routine tasks.

Furthermore, the use of AI is not only associated with workflow optimization but also with deepening knowledge and expanding programmers' skills. Generative AI tools are used as sources of real-time explanations and solutions, facilitating training in new technologies and methods. It has been statistically confirmed that generative AI is directly linked to knowledge expansion ($r = 0.580$, Sig. 0.01), emphasizing its role as a learning tool.

Breaking down the benefits of AI into cognitive, practical, and emotional categories, the results show that the primary benefit is practical — increasing efficiency and productivity (88.9%). Cognitive benefits related to knowledge expansion and critical thinking were also reported as significant (55.6%). However, emotional benefits, such as increased motivation and reduced stress, were less pronounced (16.7%). This data supports the claim that the main strength of generative AI lies in its ability to support the work and learning of programmers, but it is not yet a tool that dramatically changes their emotional state.

An intriguing conclusion can be drawn when considering the relationship between programmers' experience and their perception of AI benefits. For example, as experience increases, cognitive and emotional benefits grow, while practical benefits decrease. This can be explained by the fact that more experienced professionals have likely already developed effective work habits and techniques, limiting the impact of AI on their practical productivity. For them, AI plays a more important role as a tool for expanding knowledge and maintaining critical thinking rather than as a means of automation.

Generative AI directly competes with established professional online forums like Stack Overflow.

The study found that IT professionals use AI and online forums almost equally (M = 2.21 out of 3). While there is no significant difference in the effectiveness ratings of the two tools, generative AI has an advantage in the speed of finding solutions and the accuracy of the proposed answers. This is a significant indicator, as speed and accuracy are critical factors in the work of programmers. This advantage of AI likely explains the higher satisfaction of respondents with its use (M = 1.76 out of 3), as well as the perception of AI as a better tool for scaling their work.

While the results demonstrate significant benefits of generative AI, the study also highlights some important challenges related to its use. A key issue is the possibility that AI will automate many routine tasks, potentially reducing the demand for programmers in certain roles. The rapid development of AI technologies may require professionals to adapt to new concepts and tools, while also facing ethical questions related to the responsibility for AI-generated code and data security. Additionally, reliance on AI may reduce the creative involvement of programmers in the development process, posing a potential risk to their skill development.

Generative AI is undoubtedly changing the way software professionals work, offering significant advantages in automating routine tasks, accessing documentation, and generating code. However, as its application grows, programmers will need to adapt their skills and focus on the creative and strategic aspects of their work to remain competitive in an ever-changing technological environment.

6. Conclusion

This study highlights the significance of generative AI as a powerful tool for enhancing productivity and efficiency in programming, while also raising several questions about its long-term effects and impact on the professional identity and development of software specialists. These questions will be crucial for the future development of the industry and the role of AI within it.

One of the key findings is that generative AI is not strictly tied to the experience level of professionals. This means it offers universal value for both experienced professionals and novices, creating equal opportunities to accelerate programming skills and efficiency. However, clear usage trends are observed among different groups: practical benefits dominate, while emotional and cognitive benefits remain secondary.

These results open several important directions for future research, including: (1) the long-term effects of AI use on cognitive development and critical thinking; (2) the impact of generative AI on creativity in programming; (3) the socio-ethical aspects of AI use in programming; (4) the role of generative AI in the reskilling of IT professionals; and (5) the integration of AI with other tools and collaboration platforms.

Acknowledgements

This study is financed by the European Union-NextGenerationEU, in the frames of the National Recovery and Resilience Plan of the Republic of Bulgaria, first pillar "Innovative Bulgaria", through the Bulgarian Ministry of Education and Science (MES), Project № BG-RRP-2.004-0006-C02 "Development of research and innovation at Trakia University in service of health and sustainable well-being", subproject „Digital technologies and artificial intelligence for multimodal learning – a transgressive educational perspective for pedagogical specialists" № H001-2023.47/23.01.2024.

References:

[1]. Ebert, C., & Louridas, P. (2023). Generative AI for software practitioners. *IEEE Software*, *40*(4), 30-38. Doi: 10.1109/MS.2023.3265877

[2]. Vaswani, A., *et al.* (2017). Attention is all you need. *Advances in neural information processing systems*, Retrieved from: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf [accesed: 10 May 2024].

[3]. Ebert, C., & Louridas, P. (2023). Generative AI for software practitioners. *IEEE Software*, 40(4), 30-38. Doi: 10.1109/MS.2023.3265877

[4]. Brooks, F.P. (1995). *The Mythical Man-Month*, United States: Addison-Wesley Longman Publishing.

[5]. Budinsky, F. J., Finnie, M. A., Vlissides, J. M., & Yu, P. S. (1996). Automatic code generation from design patterns. *IBM systems Journal*, *35*(2), 151-171.

[6]. Damyanov, I., & Holmes, N. (2004). Metadata driven code generation using. NET framework. *CompSysTech*, *4*, 1-6.

[7]. France, R., & Rumpe, B. (2007). Model-driven development of complex software: A research roadmap. *Future of Software Engineering,* 37-54. IEEE. Doi: 10.1109/FOSE.2007.14

[8]. Naimi, L., Abdelmalek, H., & Jakimi, A. (2024). A DSL-based Approach for Code Generation and Navigation Process Management in a Single Page Application. *Procedia Computer Science*, *231*, 299-304. Doi: 10.1016/j.procs.2023.12.207

[9]. Bente, S., Randall, N., & Wäckerle, D. (2024). A Conceptual Framework to Transform Coding Education in Times of Generative AI. *Software Engineering im Unterricht der Hochschulen 2024*, 93-104.

[10]. Alkaissi, H., & McFarlane, S. I. (2023). Artificial hallucinations in ChatGPT: implications in scientific writing. *Cureus*, *15*(2). Doi: 10.7759/cureus.35179

[11]. Wei, J., *et al.* (2024). Long-form factuality in large language models. *ArXiv preprint, arXiv:2403.18802*.

[12]. Friedman, N. (2021). *Introducing GitHub Copilot: your AI pair programmer.* Github. Retrieved from*:* *https://github.blog/2021-06-29-introducing-github-copilot-ai-pair-programmer* [accesed: 12 May 2024].

[13]. Stack Overflow. (2024). *Stack Overflow and OpenAI Partner to Strengthen the World's Most Popular Large Language Models.* Stackoverflow. Retrieved from: https://stackoverflow.co/company/press/archive/openai-partnership [accesed: 15 May 2024].

[14]. Xu, B., *et al.* (2023). Are we ready to embrace generative AI for software Q&A?. *2023 38th IEEE/ACM International Conference on Automated Software Engineering,* 1713-1717. IEEE. Doi: 10.1109/ASE56229.2023.00023

[15]. International Trade Administration. (2024). *Bulgaria - Information and Communications Technologies.* Trade.gov. Retrieved from: https://www.trade.gov/country-commercial-guides/bulgaria-information-and-communications-technologies [accesed: 16 May 2024].

[16]. Economic Bg. (2023). *The Bulgarian IT industry in 2023: The wind of change and the recurring stumbling block.* Economic.bg. Retrieved from: https://www.economic.bg/en/a/view/the-bulgarian-it-industry-in-2023-the-wind-of-change-and-the-recurring-stumbling-block [accesed: 20 May 2024].

[17]. Sauvola, J., Tarkoma, S., Klemettinen, M., Riekki, J., & Doermann, D. (2024). Future of software development with generative AI. *Automated Software Engineering*, *31*(1), 26. Doi: 10.1007/s10515-024-00426-z

[18]. Wermelinger, M. (2023, March). Using github copilot to solve simple programming problems. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education, 1*, 172-178. Doi: 10.1145/3545945.3569830

[19]. Dakhel, A. M., *et al.* (2023). Github copilot ai pair programmer: Asset or liability?. *Journal of Systems and Software*, *203*, 111734. Doi: 10.1016/j.jss.2023.111734

[20]. Kuhail, M. A., Mathew, S. S., Khalil, A., Berengueres, J., & Shah, S. J. H. (2024). "Will I be replaced?" Assessing ChatGPT's effect on software development and programmer perceptions of AI tools. *Science of Computer Programming*, *235*, 103111. Doi: 10.1016/j.scico.2024.103111