Development of AI Tools in the Software Development—Modern Demands and Barriers

Denis Pashchenko

Independent Researcher in Software Development, Moscow, Russia

* Corresponding author. Tel.: +7-92614567495; email: denpas@rambler.ru
Manuscript submitted August 19, 2025; revised September22, 2025; accepted November 7, 2025; published November 27, 2025.
doi: 10.17706/jsw.20.2.84-94

Abstract: This research is dedicated to the process and the best practices of the formalization of artificial intelligence (AI) tools usage in software engineering in the end of 2024. The growing experience of first pioneers in this direction in 2022–2023 and visible demand on the market defined the actual business need: study the new intents and initiatives in AI usage in software development with the focus on the corresponding best practices. AI usage in software development has a lot of threats and risks (from labor and economical aspects to the information security) and the main goal of this research is to gather the opinions and visions of senior software engineers and managers into solid industry approach, that might help in future management of practical usage of AI in IT business and production processes. Pan-European research covers the experience of 27 teams in IT companies, system integrators and banks with strong in-house software development.

Keywords: artificial intelligence, software engineering, large learning model, change management

1. Introduction

The rapid expansion of Artificial Intelligence (AI) technologies is closely connected to the emergence and adoption of specialized computing hardware. Although such hardware originally appeared in the early 2000s, it only began to be actively utilized for neural network development in the 2010s. During this period, a significant acceleration occurred in the advancement of neural network–based AI systems enabled by Graphics Processing Units (GPUs) [1]. It became evident that GPUs are capable of performing large-scale parallel vector-matrix calculations efficiently, which are fundamental to modern neural network operations. This technological capability marked the start of the contemporary phase of intensive AI development.

Despite certain limitations—such as rising hardware costs and challenges associated with energy efficiency—the development of AI computing infrastructures has become one of the most influential and capital-intensive directions in the global IT sector since 2022. Consistent with typical patterns of technological expansion, AI-related tools and applications are evolving rapidly and are already exerting a notable impact on digital economic activities [2]. Major international technology corporations, including Google, Apple, and Microsoft, have invested billions of dollars into advancing AI technologies and are now integrating them into their software development pipelines. Moreover, the diffusion of AI continues to

reshape not only digital industries but also traditional economic sectors.

As with other forms of technological innovation, this process is not uniform or instantaneous. Nevertheless, according to the author of this study, the adoption of AI will be a key determinant of IT industry competitiveness and will significantly transform the traditional role of software engineers within the next decade. The diffusion of innovation, extensively examined throughout the twentieth century, often follows the model proposed by Everett Rogers, which categorizes adopters into innovators, early adopters, early majority, late majority, and laggards [3]. Research conducted in 2023 [4] demonstrated that innovators began actively implementing AI-based tools in 2022–2023, while early adopters also expressed readiness to proceed. The present study investigates the state of this adoption process in Europe at the end of 2024.

Additionally, the study assesses the risks and barriers associated with integrating AI into Software Development (SD). It is important to emphasize that emerging regulatory measures governing the use of AI in economic contexts within the European Union [5], as well as growing ethical, legal, and social concerns, are increasingly influencing the formation of best practices for the effective and responsible use of AI in software engineering.

2. The Goal, Methods and Details of the Study

The Pan-European study had been conducted in the end of 2024 (October–December) and covers the experience of 27 teams in IT companies or banks with strong in-house software development: from Russia and Ukraine to UK and Spain. The teams are representing different kinds of software developments business:

- Independent software vendors, including in-house product development like Google, Yandex, Finastra, Finshape, Deutsche bank, Sber, VTB;
- Custom development and outsourcing software services like First Line Software and Epam;
- Other IT institutions like Auxo, Arm and Dassault.

The goal of this research is to gather the opinions and visions of senior software engineers and managers into solid industry approach, that might help in future management of practical usage of AI in IT business and production processes. The study is devoted to the issue of active formalization of the use of AI-tools in the software engineering in 2022–2024 and continue the research of 2023 [4]. The hypothesis of the author in this study: “Demand of AI tools usage in software engineering is speedily rising according to innovation’s diffusing theory of Everett Rodger, and enough matured for definition of the best practices in the practical usage. Despite existing barriers such as ethical considerations, governmental regulations, and corporate policies, the industry's demand continues to grow”. The study has been conducted via Google.Forms instrument with questionnaire form and video interviews. The transition from the original data (survey responses, interview transcripts) to the presented percentage indicators was performed automatically by the Google.Forms system when entering the aggregate updated raw data into the final survey forms. All structured results have been sent in November–December, 2024 to experts to give them an opportunity to comment it before study results are finalized. Private remarks from experts had been used in study result’s finalizing.

The following research methods were used:

1) The principles of generalization and deduction to obtain consistent results in the context of active adaptation to innovations;
2) The holistic methodology to generalize the results in companies where the active stage of implementing AI tools has not yet begun;
3) The Pareto principle and Occam’s razor to find the most popular tools and approaches to using AI tools in software engineering;

4) The method of distributing innovations according to Evert Rogers to determine the nature of innovation development in the future.

The expert's panel might be described in following Table 1. There are three main characteristics—level of professional experience (in years), region in Europe and type of IT business. Such kind of characteristics are expressing the variability of software development experience in expert's panel.

Table 1. Experts in the research

No.	The Characteristic	Representation in the Expert's Study		
1	Level of professional experience in software engineering	Less than 10 years 22%	10–15 years 19%	More than 15 years 59%
2	Region of described the experience in usage of AI-tools	North and Western Europe (UK, Spain, Sweden, France, Germany, Swiss, etc.) 11%	Central and South Europe (Poland, Czechia, Hungary, Serbia Bulgaria, Cyprus, etc.) 33%	Eastern Europe and CIS (Ukraine, Russia, Armenia, Turkey Kazakhstan, Georgia, etc.) 56%
3	Types of IT-business	Independent software vendors, including in-house product development 56%	Custom development and outsourcing software services 33%	Other type of IT business 11%

As it's clear from the Table 1 that this Pan-European study covers the main geographical regions and let the senior engineers and managers take part in construction of the new vision of AI-tools using in software development. Main results of the study are described in following section of the article.

The proposed sample of 27 teams allows for representative conclusions about the European software development market, as it covers the region's leading countries from Russia and Ukraine to Spain and UK and European's largest corporations and banks: Raiffeisen Bank, Sber, Yandex, VTB, T-Bank, Auxo, 1C, Google, Finastra, Finshape, Dassault, Arm, and Deutsche Bank. The teams participating in the research work on software products used daily by more than 200 million Europeans.

3. Main Results of the Study

This initial study is structured into three integral sections, each contributing valuable insights into the utilization of AI-tools in software development companies. This particular article is describing the results only from first and second sections. The first section is devoted to the examination of the prevailing practices of AI-tools current usage within these companies. These tools, often made through Large Language Models (LLMs), rely on in-built artificial neural networks. These LLMs undergo training using self-supervised and semi-supervised learning techniques and typically comprise billions of weights [6]. Their training process is facilitated through specialized AI accelerator hardware, enabling parallel processing and utilizing vast amounts of software code from various sources, the majority of which are accessible openly. As of 2024, the software engineering domain comprises an extensive array of approximately 70 distinctive tools based on LLMs, with notable mentions being ChatGPT, Hithub Copilot, and Google Gemini.

The second section of this study delves into the common challenges that arise during the implementation of LLMs in real-world software development practices, including the formalization of their usage at the official corporate level. The third and final section is dedicated to presenting the forecasted trajectory of AI-tools in the software engineering domain. This section offers a comprehensive forecast regarding the technological future of AI-tool development in software engineering. Results of 3D sections are not described in this article.

With this clarification, the study's structure is now fully represented, encompassing all three key sections, each contributing substantively to the understanding of AI-tool deployment and its potential impact in the

software development landscape. The collective benefits derived from the use of AI tools have led to an enhanced cross-functionality among engineers, empowering them to tackle entire classes of straightforward tasks autonomously. In this context, the advent of AI in software engineering is poised to impact various levels of the IT industry's organizational and production paradigms, namely:

- Corporate level: with the introduction of AI, core business processes are poised to undergo transformation.
- Project level: roles and areas of responsibility within projects are subject to change.
- Personal level: software developers are compelled to update their skill sets to effectively collaborate with AI-based tools.

The study shows that significant share (35%) of teams and organizations already has started the implementation of AI-tools in real software development (in different process areas like coding, testing, etc.) and it's planned to do in near future for 48% of teams. Personally around 37% of experts are using AI tools in their regular job with the high frequency and it has a strong impact on their personal work in software development projects. And 30% of experts estimated the impact of the AI-tools on their professional work as average and valuable for some particular tasks in SD projects. It means that more than $\frac{2}{3}$ of experts from this study are using AI tools in SD projects regularly.

Expert panel estimated the main advantages of AI/LLMs usage in software development in the real practice of SD projects:

- Speed up of operations in the team/organization—70% of experts;
- Automation of routine operations and time saving—66% of experts;
- Software product excellence, incl. software quality, user experience and documentation—18% of experts.

By the end of 2024 the AI-tools (like Watson or GPTChat) are slowly rising in demand in SD projects for business and system analysis. Still more than 55% of experts aren't using it in system and business analysis, from the other hand experts confirmed that AI-tools are in demand for two types of tasks in this area:

1. In constant working with requirements and specifications—22% of experts;
2. Business modeling and data analysis—14% of experts.

Another research of 2024 [7] demonstrated, that even GPTChat version 3.5 (released in 2022) may easily build almost all documentation on the stage of business and system analysis. It means that in 2024 this lack of usage of AI tools in activities of analysis might be a potential competition advantage for software teams.

Current study shows much more successful practice of AI-tools usage in software design and construction. There are most popular types of tasks for LLMs in this process area:

- Coding (incl. unit-tests, stored procedures, etc.) —63% of experts;
- Fast software prototyping—33% of experts;
- Code reviews (incl. code optimization and refactoring)—26% of experts;

According to this research and some others [2, 4, 7] the work with code is the most popular propose for LLMs usage. In area of software quality assurance, the demand was rising and also aimed on the working with the code. There are most popular types of tasks for LLMs in software quality management activities:

- Writing the auto-tests—around 33% of experts;
- Managing the defects and reports analysis (30%);
- Searching of the errors and vulnerabilities in the code (15%).

Around 52% of experts don't use AI-tools in software quality management. Based on this study and also based on [7, 10] this process area might be enhanced by LLMs tools and now we may see the relevant examples of AI using in software quality assurance on the market.

Also, LLMs are useful in some other SD project activities:

- Software documentation (the manuals, video instructions, etc.)—around 48% of experts;
- Software product support (the issue tickets, help for users, etc.)—18% of experts.

For around 15% of teams there is a demand of AI-tools in the project management activities.

Expert panel estimated the value and the role of LLMs in learning and in the excellence of the software development skills in the end of 2024:

- The impact of usage LLMs in professional learning is very high—41% of experts;
- It's just one more useful tool on the board—44% of experts.

Study shows the rising of the value of LLMs in the professional learning in IT domain in comparing with results of Ref. [4].

Study confirmed that the implementation of LLMs is still a potential competition opportunity for IT companies, and the speed of implementing this innovation is very high. About 26% experts noticed, that their companies are executing the corporate plan to implement LLMs in software development. And around 22% of teams/IT organizations are in the process of the discussion about its centralized implementation and let the process go on the level of the separated teams/individuals.

Implementation of LLMs has risks of any kind: from ethical doubts to industry's regulation [8]. Problem of AI government regulation is actual and it's actively lasting more than 20 years [9]. Especially it's actual for European Union, who is well-known for its early regulating of hi-tech domains [10]. Moreover, even the process of implementation of current AI tools into software development in IT industry has their specific features. Expert's panel figured the main barriers in implementation of AI-instruments in software development in their teams and organizations in 2024:

- High level of different risks—from legal aspects to ethical—44% of experts;
- Lack of the resources (money, time, knowledge, HR-capital)—30% of experts;
- Organizational resistance of engineers and managers—18% of experts;

In comparing with Ref. [4], the level of organizational resistance is much less, but regulation of AI created additional barriers [8]. Lucky for the industry that any government's regulation in IT domain have very clear restrictions in its impact and software development domain in Europe have their "best tricks" to overcome it.

As it was mentioned before, more than 40% of teams are in the active process of AI-tools implementation in software engineering practices (with or without common centralized corporate plan) and around 18% of teams has no value in its implementation in 2024. Moreover, in 4% of teams there is a strong negative attitude to AI-instruments in software development (including its official prohibition). In 30% of teams there is an official plan or corporate policy with official recommendations how to use AI-instruments in SD projects, and in 22% of teams its usage is continuing now in test mode.

Expert's panel estimated (from the personal point of view) the advisability of the centralized corporate investments in the improvement of AI/LLMs usage in 2024:

- Might be useful, but it needs more preparation (R&D project, economical model, etc.)—37% of experts;
- It's the best time to do it (ahead of the competition)—34% of experts;
- Too early, better to start from individual and team's usage before centralized investments—26% of experts.

This set of opinions is clearly corresponding with mentioned before theory of innovation's diffusion by E. Rodgers [3] and looks very similar with results above about starting the process of AI implementation in software development. For sure, successful usage of AI tools in software engineering is still depends on the quality of those tools. Experts defined the main expectation of their teams from special AI tools in software engineering, that making AI tools more valuable for industry:

- Much more integrations with current software engineering tools (70%);

- AI tools should be much clever and easy to use (41%);
- Waiting for methodology for AI usage (33%).

Integrated environments for software engineering with significant impact of AI seems like a new horizon in IT domain. Almost all vendors of SDE are implementing elements of AI into their software tools: from Apple with X-code platform and Google with Android Studio to JetBrains.

In resuming this section of the article, it should be noticed, that trend of the usage of the LLMs in software development is actual for European region. The study confirmed the initial author's hypothesis: demand of AI tools usage in software engineering is speedily rising according to innovation's diffusing theory of Everett Rodger, and enough matured for definition of the best practices in the practical usage. Despite existing barriers such as ethical considerations, governmental regulations, and corporate policies, the industry's demand continues to grow. Current AI-instruments usage is focused on the working with the software code (in different kind of ways) and with SD project/product documentation. There is a big potential in LLMs usage in the software engineering and the lack of the centralized efforts on the corporate level might lead to the missing of the competition advantage in the software development.

Implementation of the LLMs in the real corporate software development practices is going in normal way according to Everett Rogers categorization from well-known theory of "Diffusion of Innovations" [3]:

- There are formed categories with the "innovators" and "early adopters" and it's clear: the formalization of AI-tools usage is started (the teams are executing the corporate plan to implement LLMs in software development and started its centralized management via corporate policy or recommendations);
- There is process of formation of "an early majority" category—those teams are interested in following learning of AI-tools via different ways (R&D projects, individual / team experiment, etc.).

This study confirmed the previous results from Ref. [4]: innovator's and early adopter's groups are formed in software development domain. They had started the centralized and formal process of LLMs implementation. Current AI-instruments usage is focused on the working with the software code (in different kind of ways), quality assurance and with SD project / product documentation. There is a big potential in LLMs usage in software engineering in area of system and business analysis. For some IT companies the lack of centralized efforts on the corporate level might lead to the missing of the competition advantage in the software production, connected with AI tools.

In comparing with 2023 [4, 11] we may see the rising demand of AI tools usage in 2024 in absolutely all process areas: from software design and construction to project management activities. It leads to clear conclusions about best practices:

1. Focus AI usage in software engineering on main process areas in production, start with all activities where software code and product documentations are the expected result;
2. Formalization of the process of the implementation of AI into software engineering is a key factor in overcoming all barriers;
3. Rising of interest to AI from industry and governments lead to the need of complex risk management in internal projects of implementation of AI;
4. Software engineers need new skills in AI-human interaction: as earlier you will start educate them as more effective they will be in transformation of your IT company.

Taking into account the most relevant risks and barriers in implementation of AI tools, defined in this study, the best recommendation is to start the formal project of AI implementation in software development company with RnD stage and formal project and risk management approaches. Moreover, usage of AI technologies became the new part of competition advantage and might be added as an element in modern model of perspective type of SD company [12].

4. Discussion and Conclusion

This study provides a comprehensive examination of the current landscape, implementation challenges, and future development potential of AI- and LLM-based tools in software engineering. It analyzes the present degree of adoption of AI tools, the barriers encountered during their integration into software development workflows, and offers a forward-looking assessment of their projected evolution and influence within the industry. In comparison with the early 2020s [13], the demand for LLM technologies has grown exponentially and now extends across nearly all domains of the software engineering process.

The study confirms the author's initial hypothesis: an increasing number of European software engineers are employing AI tools to support daily tasks on software projects (individual usage levels reach approximately 67%). Furthermore, innovative IT companies have already initiated systematic organizational changes aimed at standardizing and institutionalizing the use of AI tools—28% at the formal corporate level and approximately 40% at the team-based implementation level.

The findings highlight the transformative influence of LLMs across three key organizational layers:

- Corporate Level: AI tools are reshaping core business processes by increasing operational efficiency, fostering innovation, and accelerating the shift toward AI-driven workflows.
- Project Level: The integration of AI is altering established task structures and team roles, requiring adjustments in collaboration models and internal coordination.
- Individual Level: Software engineers must acquire new skillsets to engage effectively with AI tools, emphasizing the importance of professional development and continuous learning.

The operational advantages of LLMs are most evident in process areas where code and documentation are primary deliverables. Experts noted particularly strong benefits in software design, implementation, and quality assurance. However, the application of AI in system and business analysis remains underdeveloped. Less than half of the surveyed teams make use of AI tools for requirements management (22%) and business modeling (14%), indicating a substantial untapped potential for competitive advantage.

Expert assessment revealed the following primary benefits associated with AI adoption in software development:

1. Acceleration of operational performance: 70% of respondents observed significant gains in organizational and team productivity.
2. Automation of routine work: 66% reported time savings as repetitive tasks were delegated to AI tools.
3. Improved software quality: 18% identified enhancements in product outcomes, including user experience and documentation quality.

Despite these benefits, several barriers hinder broader adoption. The most commonly cited challenges include legal and ethical concerns (44%), resource and cost limitations (30%), and organizational resistance to change (18%). Moreover, while interest in AI tools is growing, over half of experts in quality assurance and systems analysis report limited use in their domains, indicating that these areas have not yet fully embraced AI's potential.

The results support the hypothesis that the diffusion of AI tools aligns with Everett Rogers' diffusion of innovations model. Clear categories of innovators and early adopters are identifiable, with emerging organizational strategies and structured implementation initiatives. Meanwhile, the early majority is beginning to engage with AI tools through exploratory use, controlled experimentation, and step-by-step learning.

Building on these insights, the study proposes several best practices in implementation of AI tools:

1. Prioritize AI adoption in core production areas like coding, testing and documentation.
2. Formalize AI implementation processes: structured approaches of implementation and formal corporate policy of its usage.

3. Develop robust risk management strategies: ethical, legal, and organizational risks must be proactively addressed to facilitate sustainable implementation.
4. Invest in skill development for engineers: training initiatives aimed at enhancing AI-human collaboration are essential to maximize the effectiveness of AI adoption.

Prioritizing AI adoption in core production areas is a simple practice. Early adoption should target the most tangible, measurable, and high-impact activities in software engineering, specifically areas where code, documentation, and test artifacts are the natural outputs. These include software design and construction, quality assurance, and software project documentation:

- Software Design and Construction: LLMs are widely used to accelerate coding tasks, generate software prototypes, and propose architectural alternatives. By directing AI integration here, teams achieve immediate time savings and reduce error-prone manual work.
- Quality Assurance: Automated test case generation, bug localization, and code analysis represent high-value applications where AI adoption can drastically reduce development cycle time. In particular, AI-based test data creation and regression testing help organizations cut a lot of repetitive manual QA activities.
- Documentation and Knowledge Management: AI tools can automatically generate developer documentation, API descriptions, and user manuals directly from code or system models. This not only accelerates delivery but also ensures consistency—a big challenge in large distributed teams.

Ad hoc or experimental use of AI tools—though common in early phases—cannot scale into enterprise-wide adoption without structured, well-defined frameworks. Formalization ensures that adoption is aligned with corporate governance, compliance, and quality standards. That's why formalizing AI implementation processes is a best practice, recommended in this research. Key components of this process include:

- Management Frameworks: establish roles and responsibilities for AI usage within the organization. Define policies for what tasks may or may not be delegated to AI in software development, especially in regulated industries (e.g., finance, healthcare, etc).
- Standardized Workflows: introduce templates and incorporate AI use norms in software development process areas. For example, define how generated code must undergo peer review, or how AI-generated docs should be validated against stakeholder needs.
- Integration with Toolchains: Embed AI into existing CI/CD and DevOps pipelines. By treating AI tools as additional services within the toolchain—rather than separate experiments—organizations can monitor performance, track improvements, and ensure reproducibility.

This study identified ethical, legal, and organizational risks as key adoption barriers. These risks must be actively managed through proactive formal planning and operational reactions according to corresponding mitigation plans united in the common risk management strategies.

- Ethical and Legal Considerations: AI-generated code may unintentionally (and mostly theoretically) replicate licensed or copyrighted content. Organizations must adopt compliance tools that verify license compatibility and ensure intellectual property safety. Additionally, policies must address data privacy when LLMs process sensitive inputs.
- Operational Risks: teams risk overreliance on AI without sufficient validation. To mitigate this, human-in-the-loop validation should be embedded into workflows. For example, code generated by AI must pass through mandatory peer review and automated testing before integration.
- Change Management: resistance to AI adoption is a challenge, confirmed in this research. Teams may fear job loss or skill devaluation. Transparent communication and active involvement of engineers in defining AI workflows are essential to avoid cultural friction.

The last very valuable best practice is the investment in skill development for engineers. The full potential

of AI integration depends not only on the technology but also on the capabilities of human engineers to collaborate effectively with AI systems.

- New Competencies: Engineers must learn how to prompt, validate, and refine AI outputs. Skills such as "prompt engineering," evaluating probabilistic outputs, and understanding LLMs limitations are becoming as critical as coding languages themselves.
- Training Programs: Organizations should implement structured upskilling initiatives—from internal workshops on AI-assisted development to partnerships with universities offering courses on applied machine learning for software engineering.
- Cross-functional Collaboration: AI adoption blurs traditional role boundaries. For example, requirements analysts may need to collaborate more closely with developers when using AI to generate specifications. This demands training not only in tools but also in communication and agile collaboration methods.

While other brilliant engineers are creating their own "virtual software developer" [14] the author of this study is sure that close future is not in "replacing humans by AI", but—in their collaboration and integration in new software engineering process standards. This study highlights the significant, albeit measured, potential of Large Language Models (LLMs) to influence software engineering practices. While current trends indicate a growing adoption of AI tools across various process areas—from software design and construction to project management—this integration is far from straightforward. The successful deployment of AI tools requires a pragmatic, business-oriented approach that carefully weighs both the potential benefits and inherent risks. At the same time, these topics are not exhausted for further research. Promising areas for development in this study in 2026 include:

1) Changes in software development teams due to the implementation of AI technologies;
2) The growing influence of AI tools in areas where their potential remains untapped, such as requirements management, the creation of infrastructure and environments for software development, and the automation of DevOps practices;
3) Changes in developer skills and manager expectations as AI tools replace manual labor and generate code and tests.

The findings of this study underline a rising interest in AI tools as a potential differentiator in the competitive IT industry [12]. However, their integration should not be viewed as a guaranteed transformative change [15] but rather as a strategic endeavor requiring meticulous planning. The rapid pace of AI development and its capabilities necessitate structured, incremental adoption strategies that prioritize return on investment and risk mitigation [16]. Businesses must focus on pilot initiatives, including R&D efforts and cost-benefit analyses, to establish scalable and sustainable frameworks for AI implementation. This research positions the software engineering sector at a pivotal moment, where cautious optimism is warranted. The adoption of AI tools can enhance productivity and improve software quality, but achieving tangible benefits demands cross-functional collaboration, targeted skill development, and a willingness to experiment. Companies that view AI tools as a complementary resource—rather than a wholesale solution—are more likely to achieve incremental gains without overextending their resources.

Moreover, the study underscores the necessity of ongoing research and development to address existing limitations and identify high-value applications for AI tools. A deliberate focus on refining tools for specific tasks, alongside careful navigation of regulatory, ethical, and operational challenges, will be critical in realizing their long-term business value. In summary, while the potential of AI tools in software engineering is evident, their adoption must be pursued with a strong emphasis on business outcomes and operational feasibility. Organizations that approach this shift with disciplined planning and measured expectations will be better positioned to capitalize on the evolving capabilities of AI tools. By doing so, they can enhance their

competitive standing without overexposing themselves to the uncertainties that accompany emerging technologies.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] Baruffaldi, S., *et al.* (2020). Identifying and measuring developments in artificial intelligence: Making the impossible possible, *OECD Science, Technology and Industry Working Papers*, 2020(05). doi: 10.1787/5f65ff7e-en

[2] Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot, arXiv Preprint, arXiv:2302.06590.

[3] Rogers, E. (2003). *Diffusion of Innovations* (5th ed). New York: Simon and Schuster.

[4] Pashchenko, D. S. (2023). Early formalization of AI-tools usage in software engineering in Europe: Study of 2023, *International Journal of Information Technology and Computer Science (IJITCS)*, 15(6): 29–36.

[5] The EU Artificial Intelligence Act. Up-to-date developments and analyses of the EU AI Act. [Online]. Available: https://artificialintelligenceact.eu/

[6] Kästner, C., Kang, E. (2020). Teaching software engineering for AI-enabled systems. *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training* (pp. 45–48). doi: 10.1145/3377814.3381714

[7] Pashchenko, D. S. (2024). Implementation of software development projects using artificial intelligence tools, *Project and Program Management*, 4: 284–295.

[8] Morley, J., Kinsey, L., Elhalal, A., *et al*. (2023). Operationalizing AI ethics: Barriers, enablers and next steps, *AI & Soc*, 38(1): 411–423. https://doi.org/10.1007/s00146-021-01308-8

[9] Yankovskiy. R., Bardov. I., & Nikiforov. A. (2023). Three legal views of a software: Source code, derivative work, and work for hire, *Derivative Work, and Work for Hire*. doi: 10.2139/ssrn.4425817

[10] European Commission. (August 2024). AI pact. [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/ai-pact

[11] Vergadia, P. (2023). AI in software development: What you need to know. *Google Clouds*. [Online]. Available: https://cloud.google.com/blog/products/ai-machine-learning/how-ai-impacts-software-development

[12] Pashchenko, D. S. (2024). Complicating the Innovative model of a high-tech IT company. *Information Technologies*, 30(12): 646–657. doi: 10.17587/it.30.646-657

[13] Barenkamp, M., Rebstadt, J., & Thomas, O. (2020). Applications of AI in classical software engineering. *AI Perspectives*, 2(1). https://doi.org/10.1186/s42467-020-00005-4

[14] Irwin, K. (2024). This software engineer AI can train other AIs, code websites by itself. *PC Mag*. [Online]. Available: https://www.pcmag.com/news/this-software-engineer-ai-can-train-other-ais-code-websites-by-itself

[15] Akhmedova, M. R., & Perova, A. E. (2021). Specifics of using artificial intelligence technologies in the IT industry. *Journal of Applied Researcha*, 1(5): 17–22.

[16] Industry research. (2023). Gartner research: Top strategic technology trends for 2024: AI-augmented development. [Online]. Available: https://www.jitterbit.com/report/gartner-research-top-strategic-technology-trends-2024-ai-augmented-development/