

Using Artificial Intelligence in Software Development

Chief Assist. Prof. Mariya Armyanova PhD
University of Economics - Varna, Varna, Bulgaria
armianova@ue-varna.bg

Assoc. Prof. Yanka Aleksandrova PhD
University of Economics - Varna, Varna, Bulgaria
yaleksandrova@ue-varna.bg

Abstract

The AI use is also heavily influencing software development. The demands on software are many, including its continuous update and ever-shorter development cycles. There is a growing need for shorter development cycles, flexibility and meeting requirements without compromising software quality. AI can help meet these requirements in software development. AI supports these goals. AI tools aid software development by supporting almost all development activities. It aids testing by making it faster and more efficient. The main possibilities to support activities in the different phases of software production are presented, based on tools that are available on the market and prototypes exploring the possible AI applications. The research purpose is to study the modern capabilities of AI in the domain of software production, its application limitations and to determine its role and place in software production.

Keywords: Artificial Intelligence, Software development, Software engineering, Code generation, Requirements Gathering, Quality Control, Feedback Management

JEL Code: C61, C88

DOI: 10.56065/IJUSV-ESS/2023.12.1.167

Introduction

Artificial intelligence (AI) is constantly expanding its fields of application. Due to advances in AI and machine learning technologies in recent years, they are being used ubiquitously in all spheres of economy and public life (Aleksandrova, 2021). Taking advantage of AI positively impacts business outcomes. The AI market includes natural language processing, AI robots, autonomous devices, computer vision, and machine learning applications. Forecast analysis shows that the AI software market will tend to increase in the coming years, reaching around US\$126 billion by 2025 (Thormundsson, 2022), and according to Statista (2023), it is even suggested that the size of the AI market may reach USD 241.80 billion by the end of 2023. The market size is expected to show a CAGR of 17.30% i.e. a volume of US\$ 738.80 billion by 2030 (Statista, 2023).

The increasing market share trend of this software also illustrates the efforts that are being put into AI software development. Scientists and software companies are working with AI technologies and discovering new areas of their application and new applications in already established areas. AI and machine learning technologies are entering the field of software development, proving their usefulness, and expanding their influence in this field as well.

There is a two-way relationship between AI and software development. On the one hand, software is constantly being developed using AI technologies, and on the other hand, AI is helping the development process. In the beginning, the application of AI in the software development process was only to help generate code. AI tools now exist to support the entire software development process, from planning to maintenance. AI uses a lot of computing and data storage resources that far exceed human capabilities. Therefore, AI tools accelerate the creation of software and gradually lead to a change in the development process and the roles of developers.

The purpose of the report is to explore the applications and limitations of the introduction of AI in software development and the possibilities for its future development. The methodology used

is a desk study and a systematic analysis of the literature. Publications in Scopus and Google Scholar for the last five years were searched. The data were analyzed in the context of the study.

1. Capabilities of AI tools to support software development processes and activities

Software development is characterized by an increased degree of uncertainty, ambiguity and chaos, and for a software development project to be successful it is important that the development process is sustainable, predictable and repeatable over time, while providing conditions for creativity and teamwork. For the implementation of a successful business, for example in the field of construction and logistics, the development of digital systems is required (Petrov, et al., 2020), which places additional requirements on software production.

Software development methodology defines the set of necessary methods and tools for software development, defines guidelines for their effective joint application, as well as for the planning, management and evaluation of the development process (Filipova et al., 2017). Methodologies take a certain approach and define general design principles, methods and rules.

At the core of almost all modern software development technologies lies the international standard ISO/IEC 12207. It is aimed at the processes of the life cycle of software products. Defines a process-based approach to software development. The standard does not define a sequence of their implementation, which determines its dynamic nature and allows its adaptation to the conditions of the specific project. According to ISO/IEC 12207, the life cycle of a software product includes three groups of processes:

- Basic, to which research, development (purchase), implementation, operation and support belong;
- Ancillary – documentation, configuration management, quality management;
- Organizational – project management, project infrastructure creation, life cycle assessment and improvement, training.

AI has the potential to be used in software development as well. AI-powered tools have been developed that can support almost all of the different software development processes (Table 1). AI capabilities for information extraction, analysis and classification support development from the stage of gathering and understanding user requirements, to the stage of maintaining software systems.

AI handles text translation in various natural languages, so it can also be trained to translate formally written rules into a programming language, i.e. to generate code. Additionally, AI can process collected customer feedback and documents and extract software requirements from them. AI is also used for quality control. It allows to simulate different situations for using the software and to identify security gaps in data transmission, network security, the execution of some processing, etc. In software maintenance, AI can be used to process collected customer feedback and determine the frequency of occurrence of a problem and thus create change requirements for the next version of the product. AI also supports ancillary development processes, such as planning, monitoring and management. Based on the collected data from previous projects, it can more accurately determine the time and funds needed for the new development. AI can detect bottlenecks in project work and the occurrence of problems or delays at minimal deviation from the plan. Thus, the company's resources can be saved. In addition, the AI can provide advice on managing and executing development tasks and thus train the development team.

Table 1. Groups of code development processes and capabilities of AI tools

№	Development process groups	AI capabilities
1	Main - research, development and support	Requirements gathering and analysis Code generation Quality control Customer feedback
2	Ancillary – documentation, configuration	Document generation

№	Development process groups	AI capabilities
	management, verification and evaluation	Feedback from system testing in quality control Finding vulnerabilities in code Code optimization
3	Organizational - project management, evaluation and improvement of the HC, training	Planning, evaluation and monitoring of the development process Generating advice to improve activities across iterations It supports learning

AI supports the entire development process. Although there is still no tool that supports the entire development process, individual tools support activities in the different development phases (Figure 1).

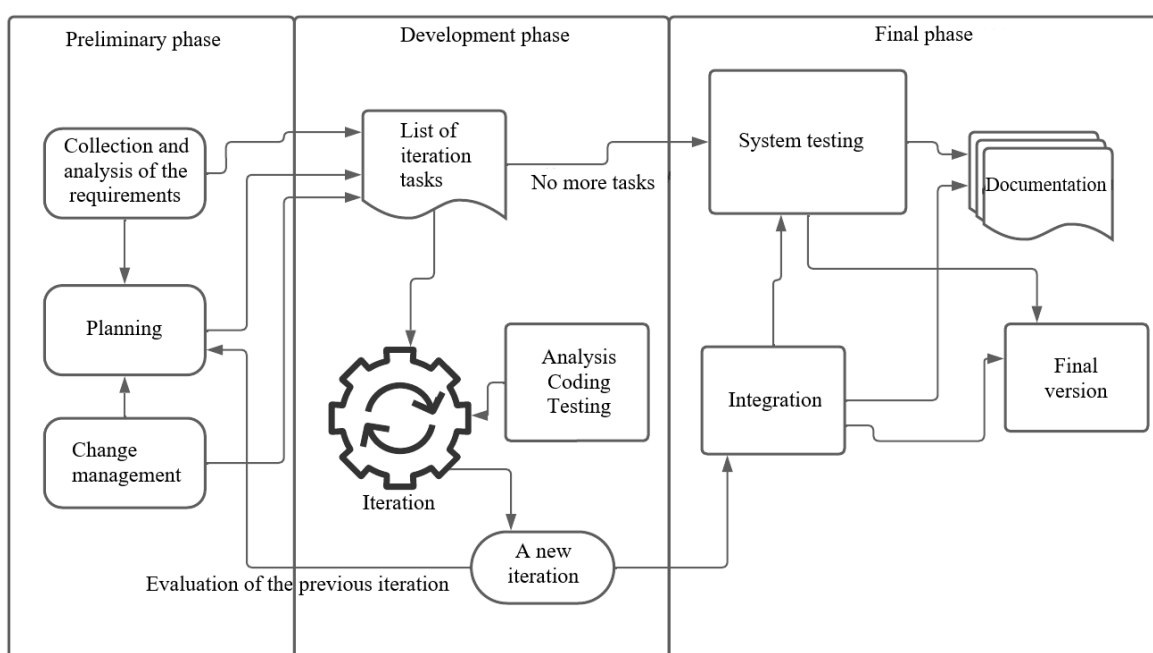


Figure 1. AI-supported software development activities

AI supports the preliminary phase by gathering software requirements and analyzing them. In addition, it has the ability to plan the budget and development deadlines, as well as monitor their implementation. AI creates a change management environment. During the development phase, it enables task prioritization, requirements analysis, code generation, and supports code testing and bug detection and optimization. After the iteration is complete, the AI can evaluate it and provide advice on how to improve the development process for the next iteration. In the final phase, AI helps with integration, system testing, and project documentation generation.

The first AI development tools are for **code generation**. Content generation AI can create code based on preset parameters. The results are satisfactory when writing small programs with a clear algorithm (Intellectsoft, 2023). But when code needs to be created for a non-standard task, or the code is required to meet certain conditions, such as speed, ease of maintenance, or modifiability, AI doesn't do so well. In general, the results are still not satisfactory and there is no substitute for an experienced programmer. However, a number of scientists (Becker&Gottschlich, 2021) are working hard and creating specialized AI in the field of generating software programs. For example, their AI Programmer software uses genetic algorithms. As a result of their efforts, training models are

becoming increasingly successful, and it is only a matter of time before AI catches up with the capabilities of junior professionals. This means that in the future, software professionals will be required not only to have coding skills, but to inspect and improve code. AI-generated code needs thorough verification to ensure its correctness, especially in terms of its security. Still, the AI Programmer prototype is specialized for a particular programming language and uses a small fraction of the language's capabilities to reduce the search area. The goal is to reduce the resources used for training and running the algorithm and thus minimize the cost of creating the prototype. Like most AI models, those in the field of code generation need more computing resources, such as memory and processing time, and so their creators use cloud resources. Even IBM's Watsonx Code Assistant for Z needs additional training and is trying to expand the programming languages it can generate code for (IBM, 2023b).

AI capabilities allow code to be generated very quickly, with the goal of reducing development time. AI can change the tasks of programmers from writing code to checking and editing code. AI is expected to create software for experienced developers to monitor and fix.

Another code development activity that AI assists with is **gathering software requirements**. AI finds application in collecting and analyzing a variety of information received through various channels for the expectations and needs of the users of the future software product. AI helps build a set of requirements for the software product. It streamlines the workflow and reduces the flow of documents related to this task that developers have to review. AI is very useful for demanding projects with complex requirements for which many documents need to be reviewed, classified and sifted. An AI product for requirements management is, for example, Watson AI developed by IBM (IBM, 2023a).

In addition, content mining products effectively assist not only in the initial research phase, but also during other stages of the software development life cycle, such as change management and software maintenance.

Content extraction AI products are aimed at business analysts who are responsible for gathering and documenting requirements, but also working with stakeholders to ensure the end product meets their needs. Extracting information from large volumes of documents in large systems is a labor-intensive, time-consuming and expensive process, and it is possible to make mistakes due to the contradiction in some documents and the peculiarities of natural language.

AI eases and accelerates the work of analysts by automating some of the tasks related to requirements gathering and documentation, and allows them to focus on end-user contacts and the selection of communication and requirements gathering channels. AI allows analysts to focus on the strategy of creating and maintaining the software.

AI tools also perform other tasks to automate the requirements gathering process. They can be used to interview prospective users of software systems, extract data from documents, and identify patterns in data. AI, based on expert systems and trained with knowledge from the professional field of software development, can assist the interviewee with difficulties in understanding the questions and suggest answers that clarify the questions. Thus, the requirements are collected more quickly and accurately.

Other AI capabilities are related to content generation. AI can also assist analysts in creating requirements documentation. Comprehensive and accurate **project documentation** can be generated based on set criteria. With properly formulated requirements, the generated documents can be clear, concise and easy to understand. The rapid generation of easy-to-understand documents allows them to be reviewed with representatives of the end user of the software and to achieve full agreement on requirements. A good understanding of requirements is key to the success of a software project. Documents are generated at every stage of software development, and this is a time-consuming and unpleasant process for most developers. AI can assist developers in generating all the necessary project documentation.

AI can be used to create shared and **secure collaborative work environments** between users and software developers. AI can follow up to confirm the understanding of the requirements from both parties and to reflect it in the project documentation in a timely manner.

There are already prototypes of AI-powered requirements gathering tools that are proving their usefulness in practice. For example, IBM's Watson AI (IBM, 2023a) is used to automate the process of gathering requirements for new software systems. IBM reports reducing the time required to gather requirements by up to 50%. At JPMorgan Chase AI (J.P. Morgan, 2023) is used to create more comprehensive and accurate requirements documentation. The use of AI leads to a reduction in the number of defects in new software systems by up to 70%. At Google (Varma, 2023), AI tools are used to support the work of analysts and provide collaboration with users. As a result, Google reported a reduction in the time needed to bring new products to market by up to 30%.

The development of AI enables the improvement of more and more system requirements analysis tasks. According to the stated guidelines for the development of existing AI tools, it is expected that the tasks of generating natural language requirements by extracting them from free-form text, such as user stories and other artifacts, are also expected to be supported in the future. Another task that can be assisted by AI is the validation of requirements against existing technologies, systems and data. Forecasts can also be created to identify and mitigate risks associated with change and inaccuracy in requirements. AI can be used to create an environment for tracking and managing changes in requirements. Another AI task related to quality control is **generating test cases** from the set requirements.

AI tools are well past the prototyping stage and tools for business analysts are emerging on the market. Therefore, specialists will have the opportunity to focus on understanding the analyzed business, communicating with future users and clients, and implementing control measures and promoting the achievement of successful project developments.

AI is extremely helpful in providing **customer feedback** in software maintenance and preparing requirements for future software release. AI has the ability to classify received signals and messages from the end user, determine the frequency of occurrence of a problem, and prioritize requirements according to their importance. It can also maintain overall communication with customers and act as an assistant, guiding customers when they encounter common difficulties and problems in using the software. In its dialogue with users, it can accumulate information about the preferred way of working of customers, the frequency of difficulties, the ease and speed of their learning and use of the software. All this can be used to generate requirements for the new version of the software.

AI can also be used for organizational activities, such as **planning, evaluating and monitoring** the development process - from making the development decision, through managing all phases of the creation and maintenance of the software product.

AI can be used in conjunction with marketing systems to **analyze the need for the software and its market feasibility**. AI is used in marketing systems for several functions (Zhang, 2021): ad programming, user experience personalization, image recognition, voice assistant, chatbots, and emotion recognition. AI also has other benefits, such as optimizing marketing efforts and reducing costs (Kniahynyckyj, 2021). AI marketing systems are used to analyze and classify the audience for full customization in ad serving. Thus, predictions can be made about consumer preferences and a set of accurate data can be determined to direct them to a specific service or product. AI allows, based on the analysis of previous campaigns, to evaluate the effectiveness of their individual elements - from the main idea to the formatting of individual elements, etc. AI helps optimize marketing efforts by relatively accurately identifying consumer behavior and uncovering the exact characteristics of likely future customers. Thus, the resources and efforts of specialists can be directed to the largest groups of customers and saved for the less likely potential customers.

Through the set requirements for the project, AI can also be used to **evaluate the economic efficiency** of the project. Based on the gathered requirements, AI can help plan and track the execution of software development activities. AI is often more effective at tasks that require predictive calculations, such as estimating costs and **scheduling deadlines**. Given enough training data, AI can produce accurate predictions by detecting dependencies that have eluded experts. In agile development methodologies that use iterative life-cycle models, data quickly accumulates about transient development iterations and their results. Analyzing the data with AI can lead to the creation of more accurate predictions and the **detection of deviations** in the performance of tasks. However, AI can also be used to generate advice for **improving the way of working** across iterations, based on the analysis of the actions of development participants in previous iterations. The use of AI can increase the maturity level of a software organization.

AI can create forecasts and determine **the composition of the development team** and the number of specialists required according to the set deadlines and the set budget.

However, AI cannot predict all new and unexpected factors, and in human resource management, such always arise. And human resource is fundamental for software development. Therefore, AI can make predictions and offer advice and solutions, but all activity is closely monitored by managers. At the current stage of the development of AI technologies, its role is only in making predictions and discovering opportunities to optimize the work process. But human resource management itself is key and cannot be outsourced to AI.

AI has a very wide application in **quality control processes** and customer feedback management. AI enables assessment of the quality of the final product (Lewowski & Madeyski, 2022). In the testing process, AI has a wide application (Pandey et al., 2021). AI can help developers by scanning for bugs and looking for vulnerabilities in the code. Efforts are underway on AI tools for vulnerability analysis (Li et al., 2019).

The most important requirements when creating software operating in the global network environment are accessibility and security (Nacheva et al., 2021). One of the strong capabilities of AI tools is to simulate the software environment and ensure security in several areas: network security, privacy security, authentication, spam filter, vulnerability and threat detection to ensure the overall functionality.

In the beginning, software professionals can train AI models using existing software security guidelines and checklists. The data for the software products created so far at the software company can then be used for training. As a result of the training, the tools can be customized and improved with the individually obtained models.

AI can be used to collect feedback from the system's usage and testing by QA specialists in a convenient automated form. A similar system can also be used for feedback to software users. AI can analyze a huge volume of reports and classify them into categories depending on the nature of the feedback (positive/negative), the severity of the reported bug (critical/major/minor), frequency of occurrence and other parameters. The resulting structured data is much more useful and easy to understand. Quality control professionals can analyze the results and identify system components that need improvement.

AI can also be used for **code optimization** (Iftikhar, 2023) by identifying code bottlenecks and suggesting more efficient solutions. For example, it can detect algorithms that are slower or use more resources. As a result, AI tools can improve performance and reduce memory usage by reducing the number of calculations or replacing some algorithms with more efficient ones.

Using generative AI tools in development teams is proving beneficial. AI contributes new ideas and concepts (Stevenson et al., 2022), for example by using chatbots to complete sentences, paragraphs or entire texts of documents based on a given context and problem definition. Code generation tools allow finding solutions with standard algorithms. Still, AI provides less original and valuable ideas than humans, but it helps team members better understand problems and

evaluate possible solutions. In several cases, AI analysis **gives confidence** especially to novice professionals and **supports their learning** by offering solutions for standard situations.

Using AI tools brings benefits to software production. They are related to speeding up the development process and ensuring predictability, sustainability, and repeatability of the development process. In this way, the maturity level of the software organization is increased. Ensures that software projects are completed on time and within budget.

Freeing managers and developers from performing routine tasks by automating many time-consuming processes allows for more creative and responsible tasks and changes their role and responsibilities.

The benefits of AI can be summarized as:

- Improve planning
- Predictable workflow and more accurate estimates
- Informed decision making
- Better security and quality control
- Reduced workload
- Speeding up production

2. Limitations of AI tools supporting software production

Despite the benefits and possibilities illustrated by the many prototypes for tools supporting software production, AI technology cannot yet replace software developers. Generative AI can generate code or generate documents, but it cannot think or behave like a software professional. According to the creators of code generation tools, at this stage of AI development, their goal is to reach the level of novice specialists (Becker & Gottschlich, 2021). The reason is that **AI cannot replace the human creativity**, intuition and domain expertise needed to develop software (Figure 2).

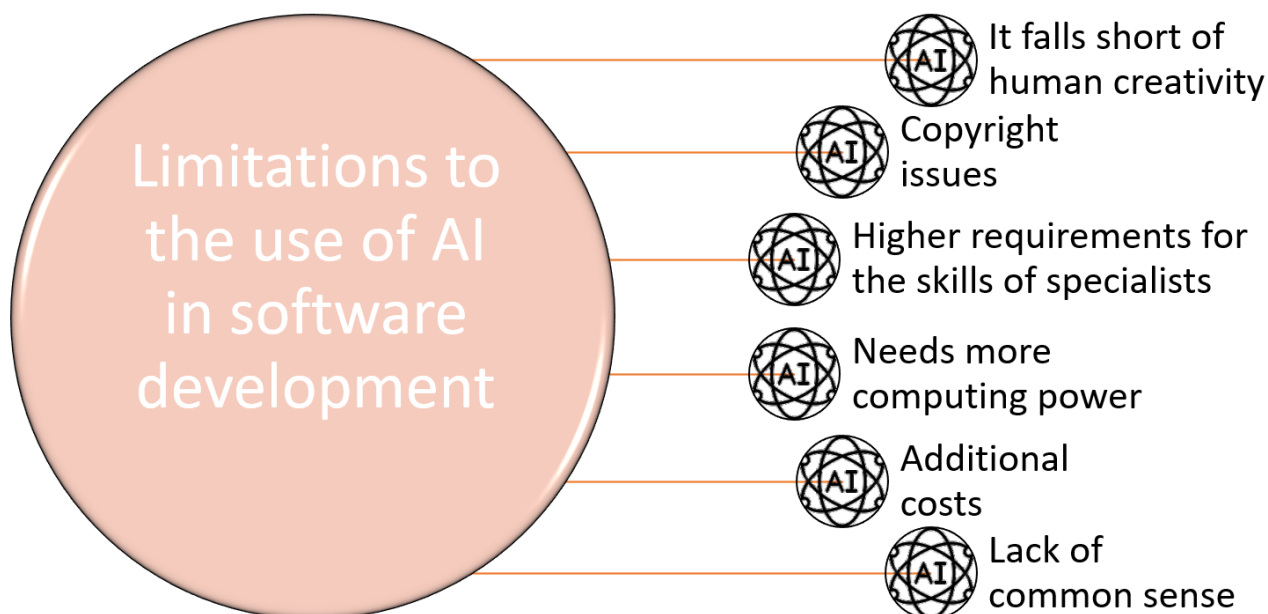


Figure 2. Restraints for using AI tools

AI's ability to generate content raises potential **copyright issues**. Although AI-generated content is not considered creativity and therefore not subject to protection, there are still quite a few unclear points, such as the use of generated content in more complex development. Since AI-

generated content does not carry copyright, this may pose some challenges for software companies to prove authorship over software code or models derived from collected software requirements. AI tools, on the other hand, are trained with large volumes of data that may be copyrighted. Some software companies prohibit the use of AI tools, especially for code generation, for fear of the possibility of infringing copyrights on the finished product or the difficulty of proving them.

Another limitation to the use of AI tools is **higher demands on the skills of specialists** (Intellectsoft, 2023). To use the potential of AI, it must be managed by a specialist with sufficient skills and experience. The reason is the requirement to formulate all input parameters with maximum accuracy. Users of the AI tools need to precisely set their requirements ie. what they want to achieve and describe it in full detail. This requires knowledge of the expected end result. They must also be able to assess how well the AI-generated solution meets the project's goals.

Advanced tools require **a lot of computing power**, storage space and power supply. A software development company that relies heavily on AI technologies must invest in top-of-the-line hardware or anticipate the cost of renting cloud resources. Better equipment means the AI can complete tasks faster. This, in turn, speeds up the work of the development team and reduces the time to create the final product.

Additional costs for acquiring a license to use the tools are also a limitation. Professional AI tools have a high license fee to use them, especially for software development. However, the costs are reasonable and justified as long as the company has the financial capabilities it needs. AI-enhanced tools quickly pay for themselves when used by skilled software professionals.

Despite the undeniable advantages of AI, there are several factors that play a restraining role in terms of its application (Bruyn, 2020) in various fields, including software production. These are **the lack of common sense** in AI, the difficulty of defining the objective function, a safe and realistic learning environment, AI bias, fuzzy model AI, the need to create the capabilities to control AI applications, and the automation paradox.

The main limitation to the penetration of AI is its lack of common sense. AI has no understanding of the basic rules that govern human existence. Any undefined goal or limitation, if not explicitly placed before the AI, does not exist and does not matter. AI applications can tackle multidimensional and complex problems that the developer hasn't even figured out cause-and-effect relationships. Therefore, AI applications can have unexpected and difficult to quantify consequences.

To overcome this AI-related risk, **human supervision** of the operation of AI tools is a must, especially when AI-supported decisions can have significant consequences for individuals and society.

In supervised learning, the objective function defines the rewards or penalties that the algorithm tries to maximize during training. Since AI does not have common sense and is not constrained by a predefined set of characteristics and model specifics, the definition of the objective function is of utmost importance. In supervised learning, it is necessary to explicitly quantify undesirable outcomes through punishments versus rewards for achieving desired outcomes.

If the model's training environment can be fully simulated, such as games with fixed rules and random possibilities, then the AI can be trained by playing against itself. However, when dealing with people, it is not possible to predict all situations. If all the factors that influence the environment are not understood and properly measured, an AI application that is tasked with gathering and analyzing user requirements will not work properly. Despite the capabilities of deep learning algorithms and large data sets, as well as the large computing capabilities of modern technologies, it is still not possible to approximate all causal relationships in a real environment. AI cannot yet reduce the need to understand causal relationships.

In some cases, the AI seems to be prejudiced. The reason lies in the data it is trained with. The statistic favors a certain factor and the model is biased towards its value. Sometimes, even if a

criterion is removed from the data, there are other criteria associated with it that there is no way to exclude from the model.

Sometimes the AI returns expected results by not evaluating the main criteria, but related ones. To be sure that the AI evaluates the important factors, it is not desirable to trust an AI with an algorithm black box (for all the reasons mentioned above). Therefore, the purpose of the smart application, the sources of the training data, and the interrelationship between the input and output data must be clear.

AI tools are **trained on historical data** from past periods, which means they are not always trained with the latest developments and technologies in the rapidly evolving fields of software development and data science. Therefore, until AI succeeds in mastering human knowledge, software professionals push the boundaries of knowledge by developing new software development tools, techniques, and paradigms. So it is difficult for AI tools to support the latest development technologies.

AI is being used to ease everyday work by replacing manual management, planning and problem solving with automatic processes. When reaching a high level of AI, it is possible that software specialists will only be expected to control the AI and intervene in emergency situations. The paradox of automation lies in the fact that it is through the execution of everyday tasks that experts are trained. At some point, there will be no well-trained experts who can correct the AI's behavior. On the other hand, it is inefficient for AI to accumulate knowledge only from knowledge bases, learning only by trial and error. Expert knowledge should be leveraged by working with AI applications, not replaced by them (Grewal et al., 2020).

Conclusion

AI tools are constantly expanding their possible applications in the field of software development. AI is a dynamically developing field in which many researchers are working and tools are being created to support all software development activities and achieve optimal results from the use of AI.

AI can help with many coding tasks and even enhance creativity, but software professionals must conceptualize software systems, understand business problems, and make strategic decisions. AI-based generation technology is not yet mature enough to replace programmers.

Despite some limitations that may have a negative role for the widespread application of AI, if approached responsibly and combining the expertise of experts with AI tools, it brings undeniable advantages to software production. Therefore, the role and capabilities of AI that support the execution of tasks in the future are expected to grow.

AI is not expected to replace developers but is likely to change their tasks. Software professionals may use AI tools for time-consuming routines. Developers may be required to solve innovative and challenging tasks. Even at its current stage of development, AI tools are accelerating software development processes and increasing software productivity and efficiency.

References

1. Aleksandrova, Y. (2021) Comparing Performance of Machine Learning Algorithms for Default Risk Prediction in Peer to Peer Lending. *TEM Journal*. vol. 10, is. 1, pp. 133-143.
2. Becker, K., Gottschlich, J. (2021) *AI Programmer: Autonomously Creating Software Programs Using Genetic Algorithms*. GECCO '21: Proceedings of the Genetic and Evolutionary Computation Conference Companion. 08 July 2021. pp.1513–1521.
3. Bruyn, A., (2020). Artificial Intelligence and Marketing: Pitfalls and Opportunities, *Journal of Interactive Marketing*. vol. 51. August 2020. pp. 91-105.
4. Filipova, N., Parusheva, S., Aleksandrova, Y. (2017) *Fundamentals of information systems*. Science and Economics. Varna.

5. Grewal, D., et al., (2020). Frontline cyborgs at your service: How human enhancement technologies affect customer experiences in retail, sales, and service settings, *Journal of Interactive Marketing*. vol. 51, pp. 9–25.
6. IBM. (2023a) *IBM Plans to Make Llama 2 Available within its Watsonx AI and Data Platform*. [Online] Available from: <https://newsroom.ibm.com/2023-08-09-IBM-Plans-to-Make-Llama-2-Available-within-its-Watsonx-AI-and-Data-Platform> [Accessed 27/11/2023].
7. IBM. (2023b) *IBM Unveils watsonx Generative AI Capabilities to Accelerate Mainframe Application Modernization*. [Online] Available from: <https://newsroom.ibm.com/2023-08-22-IBM-Unveils-watsonx-Generative-AI-Capabilities-to-Accelerate-Mainframe-Application-Modernization> [Accessed 27/11/2023].
8. Iftikhar, H. (2023) *The Role of AI in Software Engineering in 2023: Where is the Tech Industry Heading?* [Online] Available from: <https://medium.com/tech-lead-hub/the-role-of-ai-in-software-engineering-in-2023-where-is-the-tech-industry-heading-87185022dffa> [Accessed 27/11/2023].
9. Intellectsoft. (2023) *Benefits and Perspectives of Artificial Intelligence in Software Development*. [Online] Available from: <https://www.intellectsoft.net/blog/benefits-and-perspectives-of-artificial-intelligence-in-software-development/> [Accessed 27/11/2023].
10. J.P. Morgan. (2023) J.P. Morgan AI Research Awards. [Online] Available from: <https://www.jpmorgan.com/technology/artificial-intelligence/research-awards> [Accessed 27/11/2023].
11. Kniahynyckyj, R. (2021) *The pros and cons of AI in marketing*. [Online]. Available from: <https://business.twitter.com/en/blog/the-pros-cons-ai-in-marketing.html> [Accessed 12/11/2021].
12. Lewowski, T., Madeyski, L. (2022) Code smells detection using artificial intelligence techniques: A business-driven systematic review. *Developments in Information I& Knowledge Management for Business Applications*. pp.285–319.
13. Li, Z., Zou, D., Tang, J., Zhang, Z., Sun, M. Jin, H. (2019) A Comparative Study of Deep Learning-Based Vulnerability Detection System. *IEEE Access* 7, 2019. pp.103184–103197.
14. Nacheva, R., Sulova, S., Penchev, B. (2021) Where Security Meets Accessibility: Mobile Research Ecosystem. *International Conference on Electronic Governance and Open Society: Challenges in Eurasia*. Springer International Publishing. pp.216-231.
15. Pandey, S., Mishra, R, Tripathi, A. (2021) Machine learning based methods for software fault prediction: A survey, *Expert Systems with Applications* 172. p.114595.
16. Petrov, P., Sulova, S., Radev, M., Aleksandrova, Y., Mileva, L., Yankov, P. (2020) *Digitization of business processes in construction and logistics*. Knowledge and business. book 8.
17. Statista. (2023) *Artificial Intelligence – Worldwide*. [Online]. Available from: <https://www.statista.com/outlook/tmo/artificial-intelligence/worldwide> [Accessed 27/11/2023].
18. Stevenson, C., Smal, I., Baas, M., Grasman, R., van der Maas, H. (2022) Putting GPT3’s Creativity to the (Alternative Uses) Test. In *International Conference on Computational Creativity (ICCC) 2022*.
19. Thormundsson, B. (2022) *Artificial intelligence software market revenue worldwide 2018-2025*. [Online] Available from: <https://www.statista.com/statistics/607716/worldwide-artificial-intelligence-market-revenues/> [Accessed 27/11/2023].
20. Varma, K. (2023) How AI is Automating Requirement Gathering and Documentation: A Perspective from an AI Tool. Available from: <https://www.linkedin.com/pulse/how-ai-automating-requirement-gathering-documentation-kiron-varma/> [Accessed 27/11/2023].
21. Zhang, Y. (2021) *Artificial intelligence in marketing: History, definition, types, examples and benefits*. [Online]. Available from: <https://hapticmedia.com/blog/artificial-intelligence-in-marketing-definition-example-benefit/> [Accessed 02/11/2021].