AI use in Software Engineering: More than you bargained for?

Peter Bednar[1,2], Christine Welch[3]

[1] School of Computing, University of Portsmouth, UK
[2] Department of Informatics, Lund University, Sweden
[3]Faculty of Business & Law, University of Portsmouth, UK

Abstract

The paper relates to the use of Generative Artificial Intelligence in software engineering practice. The nature of Generative Artificial Intelligence is introduced. Drawing upon the literature of the field, the positive benefits of AI use are considered and discussed in relation to some of the drawbacks and challenges involved. The discussion then moves on to examine the views of professionals, ascertained through a longitudinal study involving interviews with experienced, senior software engineers from a number of different organizations. These short interviews were focused on participants' immediate experience of using GenAI software in their work as professional developers. All companies involved in the investigation have longstanding and significant IT development departments for in-house ICT projects, and client-oriented, collaborative business-to-business projects. The results suggest that an initial experience of significant productivity improvement from AI-supported software development was critically hampered by quality issues. There were unexpected and undesirable software issues and problems, requiring laborious and resource-intensive efforts to address them retrospectively. Significant delays and drains on resources resulted. Some key points for reflection are drawn from the results of this small study.

Keywords

AI supported work, Software Engineering, Sociotechnical Perspectives on GenAI use.

1. Introduction

The purpose of this paper is to consider the advantages and challenges involved in use of Generative AI tools in the field of software engineering, as an example of a sociotechnical endeavour. In the 21st Century, much progress has been made in the field of Artificial Intelligence, leading to the possibility of enhancing processes, for instance in banking and detection of crime (Ehrndal, 2025). AI that appears to mimic the ways in which human beings think has been driven by developments in neural networks and machine learning,

STPIS'25: The 11th International Workshop on Socio-Technical Perspectives in IS (STPIS'25) September 17-18 2025 Skopje, North Macedonia.

peter.bednar@port.ac.uk (P. Bednar); peter.bednar@ics.lu.se (P. Bednar); christine.welch@port.ac.uk (C.Welch)

0000-0002-3631-2626 (P. Bednar). 0000-0002-7149-3354 (C. Welch)

which have enabled natural language processing. Nowadays, it is necessary to make a distinction between what may be termed 'traditional' AI and Generative AI. The former includes predictive AI; conversational AI used in chatbots and on-line customer support; and AI-based automation, which merges into robotics and underpins Industry 4.0 systems. Generative AI, on the other hand, employs Deep Learning models to produce novel output. Much interest in finding applications for GenAI is apparent in industry and commerce.
For example, business commentators Forbes.com suggest:

"*GenAI represents a new paradigm in how software is developed, and it's revolutionizing the entire landscape of software engineering. Unlike traditional approaches that rely on human expertise and labor-intensive processes, GenAI empowers developers with intelligent tools capable of generating code, suggesting improvements and even anticipating potential issues— all in real time.*" (Graham, forbes.com, 2024)

While it is easy to recognize that there is attraction to GenAI use within the field, we believe that such enthusiasm is maybe premature. The next section of the paper will look at the nature of GenAI, the purported benefits of using it and the challenges posed by incorporation of GenAI within sociotechnical systems, taking Software Engineering as a specific example. We then go on to report the findings of a small, longitudinal study into the experiences of senior Software Engineering professionals of their experiences of working using GenAI-based support.

The Nature of Generative AI

Neural networks have underpinned the development of AI. These comprise interconnected nodes (artificial neurons), structured in layers, that process and 'learn' from data. Feedforward neural networks, in which data flows in one direction, can perform tasks such as pattern recognition, choice and classification. Their structure consists of three layers: an input layer, a hidden layer and an output layer. The hidden layer transforms the input into something useable by the output layer, using key parameters such as weights and biases. Learning can take place as parameters are updated in response to new data or conditions. Each adjustment brings about an evolution in the response of the network, so that it can adapt to different tasks or environments. Neural networks of greater complexity, comprising many layers, have enabled further developments in the AI field through Deep Learning (Hinton, Osindero & Teh, 2006). Deep learning models can use unsupervised learning to extract characteristics, features, and relationships from unstructured, unlabelled data. The multiple layers of interconnected nodes in a Deep neural network will each build on the previous layer to refine and optimize output and drive learning. The boundaries of AI have thus been moved on and appear to be closer to human reasoning through natural language processing, image processing and even a semblance to creativity. Generative AI employs Deep Learning models to produce novel output. A number of different techniques may be applied. *Generative adversarial networks* (GANs) create new data resembling the original data on which they are trained via a back-and-forth 'dialogue' between a generator and a discriminator. Another technique is *Diffusion*, which makes use of progressive Gaussian noise-addition and denoising, until original data is unrecognizable, but its patterns remain. In *Transformer models*, an encoder converts raw, unannotated text into representations known as *embeddings*. A decoder takes these embeddings, together with previous outputs of the model, and makes successive predictions.

Large Language Models (LLMs) are advanced machine learning models designed to understand and generate human language. The largest and most 'capable' LLMs are generative pretrained transformers. They are trained using vast amounts of text data, giving them the capability to perform natural language processing (NLP) tasks. While LLMs do not have agency in themselves, they are used in a variety of applications, such as generating text, translating, summarizing content, and answering questions.

2. Use of Generative Artificial Intelligence in Software Engineering

There is a great deal of interest in the literature relating to this application. Much of this focuses upon the benefits of AI use. Relatively little attention to date appears to have been given to the specifics of Generative AI.

A number of benefits are put forward for those adopting GenAI to carry out professional work, such as software development. Singh (2024) suggests that GenAI tools may afford a number of benefits, including enhanced workflow; fostering of creativity; automate bug detection; reduction in costs and democratization of Software Engineering itself. It can be seen that there is some justification for these claims. It is acknowledged that there are traditional AI tools that may be useful in managing the SE lifecycle. It may be possible to use GenAI as a means to generate and play with prototype software, stimulating creativity among software engineers; and there are tools that will find software bugs faster than a human being. However, the assertion that GenAI can democratize Software Engineering bears closer examination. Certainly, there are tools available that are relatively low cost and easy to use for a novice, thanks to natural language processing. It is possible to ask such tools to create a piece of code for a specified purpose. AI assistants are available that are usable by a beginner and will create code from natural language prompts, e.g. GitHub Copilot. This is possible because there are standard patterns of technical language that AI assistants can glean from their training data (Jaiswal, 2025). Such a process has been termed 'vibe coding' because it is driven by mere desire, rather than careful specification of requirements. Of course, this also means that it pays no regard to more demanding and possibly tedious matters, such as the need for security or recoverability if the code breaks down. AI assistants do not address such needs, but merely recognize and replicate the patterns of existing working code (Bahrini, et al, 2023; Yetiştiren, et al, 2023).

Thus, it is easy to see how the production process can be speeded up, and no particular professional skills are needed in order to become productive, but Software Engineering comprises more than just coding. If the necessary stress testing and the requirements of integration are considered, both the claim to democratization and to cost-saving are premature. As Feldt, et al (2018) pointed out, while AI use can bring benefits to Software Engineering life cycles, new functionality also brings new challenges. There is a need to focus upon the different ways in which AI may be applied in SE and the different types of risk that each may entail.

Kumar (2025) suggests benefits of AI use could include getting the product to market faster; continuous delivery; built-in security; and improvements in collaboration among technical and non-technical members of a development team. However, this author also emphasises that there is a need to focus on synergy between AI, security issues and accessibility factors.

An overview of the early literature of this field was given by Batarseh, et al (2020), who identified four 'dilemmas' in AI use. First, they highlighted a need to clarify the role of human beings in Software Development. Can the work of SE professionals be reduced to simply monitoring work performed by AI? Secondly, they point out that software can be created by AI, but AI is itself software. They question whether any development cycle exists as such when GenAI tools are used? It appears to be a closed circle. What are the implications of this for the quality of software and for the development of the profession? Thirdly, most of the evaluation research appears to focus on realisability of benefits from using AI in SE from a 'scientific' perspective. However, software engineers themselves regard it as both a science and an art. What happens to the artistic dimension if AI drives the whole life cycle? Will the evolution of software, and by extension of sociotechnical systems, become restricted and stunted. Finally, these authors highlighted that there could be inherent conflict between the development of AI agents (whose purpose is to demonstrate the possibilities of non-human 'intelligence') and the purposes of software engineering that require much more – '*to a build valid, verified system, on schedule, within cost and without any maintenance, or user acceptance issues'* (2020, p.38).

There are undoubted drawbacks to the use of GenAI in creating software, quite apart from the risks mentioned above. Any output from such tools can only ever be unreliable since their underlying logic is to generate a plausible outcome, based on pattern recognition. Gen AI assistants have no in-built conceptions of either truth or reality (see Yetiştiren, et al, 2023; Morrone, 2024), merely probability. In any application in which reliability and accuracy are crucial, this will be a major difficulty. It is interesting to reflect upon examples from another field in which 'truth' and reliability are of great importance. Difficulties have arisen with use of GenAI tools within legal systems around the world. For instance Harkess refers to the case of Ayinde v Haringey which came before the High Court in London (Ayinde v London Borough of Haringey, 2025; Harkess, 2025; Khan, 2025). The junior barrister appearing for the claimant cited five cases in support of a legal argument, including quotations from judgments. These had been found using a GenAI search tool. It transpired that the cases were non-existent, and the quotations fabricated. The cases were confabulated by the GenAI tool she had used. These 'hallucinations' of GenAI appeared completely plausible in comparison to real law reports, except that there were suspicious traces – an American spelling, for instance, which no genuine English court reporter would adopt. Having been asked for five supporting cases, the tool duly provided them, using patterns gleaned from its training data to confabulate what was not available in reality. The lawyer narrowly avoided being held in contempt of court. In her judgment, the presiding judge set out five lessons that must be learned by the legal profession for future practice: 1. that GenAI tools are fundamentally unreliable; 2. they are driven by prediction, not verification; 3. that a legal training increases vulnerability because these confabulations conform to expectations and are plausible; 4. that there is an absolute duty of verification of all sources derived from GenAI and a rebuttable presumption of AI fabrication; and 5. that consequences for improper use will be severe and inevitable. Similar judgments have arisen in other jurisdictions, such as the United States Supreme Court (Surden, 2024) and Australia (Taylor, 2025). The process of verification is estimated to require twice the time and effort of producing the output, thus negating, to some extent, the suggestion savings in both time and cost.

A further point to consider is a benefit mentioned by Alenezi and Akour (2025). They suggest that GenAI might be used to elicit and specify user requirements. According to these authors, traditional methods used to elicit requirements are labour-intensive and '*susceptible to misinterpretations and omissions*' (2025, p.6). AI Systems, they claim, can extract meaningful requirements, including both functional and non-functional elements, by processing unstructured data such as emails, minutes of meetings and feedback forms. Using natural language processing they would be able to detect ambiguities and mitigate the risk of misunderstandings. (This suggestion appears to conflate a process of elicitation for the purpose of specifying program design with a process of analysis underpinning system design). As has been pointed out elsewhere (Nissen, 2002) most professionals do not regard themselves primarily as users of software, and any such material as emails and meeting records would give only the merest glimpse into their working lives. From a sociotechnical perspective, such an approach can only be regarded as questionable, since only the engaged actors could have meaningful understanding of contextual dependencies involved in complex roles and interactions. It might, on the other hand, be possible to engage users in surfacing and making explicit their needs from the system using an intelligent assistant with which they can establish a dialogue. Users need support for their sense-making so that they can move from a situation of uncertainty to one of ambiguity. Bednar, et al (2014) describe such an agent, which can support ambiguity through application of paraconsistent logic. By extension, it may be possible to use such assistants to support meaningful dialogue among users about their system requirements.

As can be seen, GenAI tools may have much to offer in helping to improve productivity in fields such as software engineering, by supporting prototyping, improving workflow and enhancing error detection. However, proponents would do well to consider the tasks that will be involved in ensuring reliability, validity and security. These must be set against the desired productivity gains.

The next section of the paper looks at a small study designed to gauge opinions of professional software engineers regarding use of GenAI tools in their work. These snapshot accounts from practice were derived from a longitudinal study conducted between February 2023 and January 2025.

3. The study

Purpose of Study

The purpose of the study was to obtain snapshot views of real-world experiences and practice of AI use in software engineering, by exploring participants' views of professional experience within individual work contexts, from their own, unique perspectives.

Philosophical Perspective

The authors have approached this study taking an interpretive stance. Many sociotechnical inquiries are conducted from a desire to uncover generalisable principles for good practice, focusing upon precision and clarity in expressing an issue or problem. This approach has been termed logical empiricism. However, Radnitzky (1973) points to a danger that such an approach promotes an artificial separation between observers, with their unique worldviews, and the observations of phenomena that they make. This inquiry is therefore based in an alternative philosophical paradigm of hermeneutic dialectics. Here, there is explicit recognition of uncertainty and ambiguity, as features of socially constructed perspectives on human activity. Rather than precision and clarity, focus is on transparency, emphasising individual self-awareness.

Such an interpretive stance emphasises relevance in findings, rather than rigour and generalisability. The material uncovered is considered to be illuminating in its own right. Researchers adopting an LE stance will make provision in their design of inquiry for measures of reliability and validity. Within an HD-informed study, on the other hand, the appropriate measures will be transparency and recoverability. Thus, coding is used so that it will be possible to attribute particular findings to particular subjects, and the observations recorded are fed back to the participants, so that they have an opportunity both to confirm and reflect upon them.

Research Ethics

The research was conducted within the Ethical Framework adopted and published by the University of Portsmouth. This is grounded in the UKRIO Code of Practice for Research. An ethics review was conducted at the beginning of the project, taking into account the principles set out in this framework. These include, inter alia, assessing whether the study would maximise benefit for individuals and society, while minimising risk/harm to participants; respecting privacy, autonomy, diversity, values and dignity of individuals, group and communities; ensuring that participation is informed and voluntary, and can be withdrawn at any time; ensuring that activities are conducted with integrity and transparency, using appropriate methods. Full details of this framework can be found at https://policies.docstore.port.ac.uk/policy-028.pdf.

In the context of this inquiry, it was possible to involve participants in discussion of ethical considerations during the first round of interviews, including decisions on such matters as the need for anonymity. The results of these discussions were incorporate into the ongoing research design. Initially, not all of the interviewees saw a need for anonymity. However, this changed over time, and by the 3rd round of interviews everyone involved agreed that anonymity had been the right choice. This was a direct reflection of their changes in perspective, becoming more and more sensitive and potentially challenging to the viewpoints of their own managers and the corporate environment. This reflection on ethical commitment also required that the identities of their employers to be kept anonymous.

It was further agreed with the interviewees no recordings should be made. All notes were therefore made on paper, and consolidated during, as well as immediately after the meetings. Personal names, company names, places, etc were explicitly excluded from any notes.

Methodology

Recruitment and Sampling

Convenience sampling was used to recruit participants in this small study, drawing on personal recommendations and connections with professionals, including some alumni of the university. The selection criteria were specifically, [a] senior professionals with significant experience as lead software developers and project managers; [b] professionals who were actively working in organizations with relatively large scale (in house) projects, in departments, teams or groups of more than 50 software developers, contracted as employees. This resulted in 7 participants, all of whom were employed at a senior level. The interviewees were the same throughout all four interviews.

Table 1 shows the range of employing organisations and the professional role of participants, providing context for their reflections.

ID	PROFESSIONAL ROLE	Years of experience	CATEGORY OF COMPANY	Software Developers inhouse
1	Senior Software Developer	20+	Multinational Bank1	10k+
2	Senior Project Manager/ Developer	15+	Multinational Bank2	10k+
3	Senior Software Developer	20+	Large Insurance Company	2k+
4	Senior Software Lead	25+	Multinational Manufacturer	3k+
5	Senior Software Developer	15+	Medium Size Space-Tech Company	200+
6	Chief Software Developer	15+	Multinational Hi-Tech Company	500+
7	Senior Software Developer	30+	Aviation Industry	5k+

Method

Four rounds of informal, on-line interviews were conducted, using Teams, Google Meet, Zoom etc dependent on the convenience and preferences of the interviewees. The interviews were short (approximately 15 minutes), sometimes during lunch breaks, sometimes in the late afternoon (after 17:00 when support staff had left for the day). While areas for discussion were considered during the planning stage, the interviews took the form of informal, guided conversations, in which participants were invited to explore their own perceptions from professional practice, within the context of their own work environments. Open-ended, generic questions were used to initiate conversation, asking participants to talk about their experiences and personal views on the situation and use of GenAI tools and technologies. Notes were made and fed back to the participants, during and after the conclusion of each interview.

Each snapshot response was systematically and carefully evaluated, interpreted and compared with the others. Simple differences in wording were ignored, and some acceptance of vagueness was tolerated, as the main focus of this study is on potential issues with the overall experience, and change of experience, as described. Such issues have potential consequences for professionals using this kind of technology to enhance their effectivity. This is so even in areas with which they are intrinsically familiar and where they already have deep knowledge. There are also implications for academics.

Findings

The first round of interviews was conducted during the Spring of 2023. At this stage, the opportunity was taken to discuss ethical considerations. The participants indicated their view that anonymity was necessary in recording the results in the ongoing study (see Research Ethics section above).

Table 2 shows the reflections of participants on use, policies and practice of AI in software engineering during the first round of interviews.

ID	Note
1	Used AI experimentally in their work, they were enthusiastic about it. For coding, design as well as for documentation.
2	Colleagues used it more than once a week.
3	Use of AI tools was informally supported by their local management.
4	Use of AI was not formally supported by management.
5	Limited use of AI, without formal policy in place.
6	Only tentative use and experiments not related to projects.
7	Heard of it but does not know anyone using it in their project work.

As was to be expected, use of AI, and development of internal policies for its use, extended over time. While Table 2 contains brief accounts of the participants' awareness of a role for AI in software development, actual use was not reported in all cases. Some participants were able to report on management policies; in most cases use of AI was tentative and exploratory only. It should be noted that a variety of GenAI tools were in use in the different companies represented in the study. It is not intended to reflect upon the usefulness of any of these in relation to others.

Table 3 shows the responses of participants during the second round of interviews, which took place in the autumn of 2023. It can be seen that these are considerable fuller than those recorded in the first round. This may be in part because of rapid growth in awareness and use of AI tools in professional practice, but may also be in part due to development in the participants' interest in, and sensitivity to, the impact of such tools within their context of use as the study progresses.

ID	Note
1	Used AI output regularly in their work more than once a week. For coding, design as well as for documentation.
2	Use of AI was usually supported (informally) by management. Some with caution.
3	Productivity of AI use were experienced as very impressive and significant.
4	There were concerns with that there were issues and difficulties with the use of the results and inconsistencies in documentation.
5	Significant number of staff are playing with it, no specific policy.
6	Some colleagues are promoting it but some are concerned about code quality.
7	Some trial use but risks are unclear, widespread hesitation among colleagues. No corporate recommendations.

The third round of interviews took place during the Spring of 2024.

Table 4 shows the contextual reflections of participants from this round. It can be seen that, once again, the amount of material recorded from each response has expanded. Here, we can clearly see that policies for use of AI tools are beginning to be laid down by managers. There are also details of perceived problems and challenges that have arisen in context and remain to be addressed.

ID	Note
1	Used AI regularly in their work more than once a week.
2	Use of AI was actively promoted and pushed by management.
3	Use of AI was actively promoted by corporate leadership and senior management.
4	Started to experience that they had serious delays in projects due to discovery of unexpected issues and problems with the use of AI output. The scale of the productivity increases in previous period of use of AI output had introduced an unprecedented backlog and setback with regards to quality control and testing of solution.
5	Described issues were linked to large increase in smelly code, including logical bugs, feature errors and flaws introduced which took time and effort to identify and correct.
6	Issues with increase of smelly code also impacted features and design which was intrinsically difficult to validate and verify.
7	Generated documentation tended to be described as awkward, time consuming to revise and correct and did not match code on logical level.

The fourth and final round of interviews took place during the autumn of 2024.

Table 5 shows the responses of participants during conversations in round 4.

ID	Note
1	Use AI regularly more than once a week. Junior staff happy senior staff raising concerns regarding quality and robustness. Major quality concerns related to promotion of expanded use of AI by corporate leadership.
2	AI use continues to be promoted, pushed by management. Project lead concerns experienced as falling on deaf ears. Senior staff worried about security and time wasted of reverse-engineering solutions continuing having undocumented functionality.
3	AI actively promoted by corporate leadership. Project managers concerned as quality assurance issues are not addressed successfully. Projects halted and delayed due to due to issues that are difficult to identify.
4	Project managers worried about the integration and expansion of the use of AI in code generation as well as in documentation. Experience unresolved, serious delays, due to unexpected software issues and code related problems with the massive expansion of generated code. The scale of the productivity increases previously described not experienced as sustainable due to quality control issues.
5	Senior Developers experience increase of issues linked to increase use of generated smelly code, feature errors the biggest and most time-consuming to address. Top management drive for use AI felt as a concern.
6	Issues with AI code have a negative and costly impact specifically on security features. Concerns regarding leadership unreasonably positive attitude and expectation on existing AI capabilities.
7	Big issue with documentation not describing true logic of code / solution. Policy impossible to implement in practice: requires integration of use of AI technology in generation of code and documentation while requiring all to be validated. For which appropriate resources are not experienced as being provided.

The notes have again become fuller and more elaborate, and it can be seen that participants are reporting a division among their colleagues. Junior staff are appreciating the qualities of AI tools in terms of ease of use and productivity. Senior managers appear to be keen to encourage use of these tools in order to reap their purported benefits for the company. However, at the same time, senior professionals and some managers are beginning to have severe misgivings about emergent difficulties. Perceived problems appear to coalesce around quality issues. While there are undoubted productivity gains, work is often undocumented. This means that reverse engineering is required in order to trace origins of poor functionality. Concerns are raised about sustainability and security. Where work is documented, there may be a mismatch between the documentation and the actual logic of code produced. There is recognition by senior professionals that extra work must be done to validate output produced using AI tools. However, this recognition does not always include senior managers who provide resources and, in some cases, have been keen to reap the promised productivity gains. In this last round of interviews it was clear that all participants had developed strong views and reflections on the topic.

Table 6 provides a brief summary of the factors emerging from the interviews.

ID	Note
1	Issues with content and working features that were undocumented and unknown and discovered retrospectively. These were difficult to identify and added features tended to be undesirable and unintentionally introduced into solutions.
2	Issues with significant project delays. They were convinced that the work required to address the issues with the use of AI output in their work necessitated significantly more time and resources in comparison with not having used AI output in their work in the first place.
3	Issues with security in general
4	Issues with code quality in general
5	Issues with documentation in general
6	Concerns with corporate promotion and policy
7	Concerns with implementation and practice. Results getting worse, not better.

Reflections

There are a number of interesting points emerging from the various conversations. These are perceptions from senior professional software engineers, who may be considered well-informed about the context of practice within which AI tools are to be deployed. By the last round of interviews, it became clear that all participants had developed strong views and reflections on the topic.

Interviewees concluded that real-world cost and penalty of using existing AI tools went far beyond any benefit that an appearance of initial productivity gains would justify. Participants were concerned about a conflict between their own experience of use of AI tools and the ongoing insistence and promotion of use by their corporate leadership. They did not experience that they were being listened to, as promises of productivity gains and cost saving completely skewed the expectations of corporate management. Senior managers seemed to be prepared to turn a blind eye to the explosion of cost and delay experienced by those involved in practice. They failed to relate productivity gains to the extra resources required when reverse engineering became necessary, to identify and address undocumented flaws and problems. There appeared to be an ongoing mismatch of expectations and reality of experiences. The interviewed professionals did not see a constructive way out of the situation or a ready solution to these difficulties. They did not feel their concerns were listened to, or that any lessons were being learned or taken on board by corporate management.

4. Emerging Areas of Concern

As a result of the study, the authors consider that there are areas that would be worthy of deeper consideration in future research.

First, it appears that corporate management in a number of companies appeared not to be listening to senior professionals or engaging in real dialogue about deployment of GenAI tools. Ordinarily, it would be expected that the opinions of such colleagues would be relevant and influential on promotion of these or any other tools in a context of use. Promises made for GenAI, and the capabilities for which it is promoted, are very attractive within a competitive business or public service environment. Thus, wishful thinking and desire may lead people to engage with them. People who are not technical experts may become susceptible to disqualifying critical mindsets and ignoring professional experiences and misgivings. In the final stages of the study reported here, there was a clear agreement by the end that use of the tools had created more problems and issues over time, ironically while being pushed and promoted more than ever before.

There are a number of key points for consideration:

How were the tools to be made actionable?

Resources or support not available for professionals on how to apply verification and validation of AI output, and how source necessary resources to address quality and security issues.

Lack of guidance

No description or advice was available as to what kind of workarounds were acceptable. There was no guidance on what to do when necessary resources to address practical problems were not available.

Lack of dialogue:

No feedback or discussion took place regarding development of contextually-relevant practices for quality assurance.

No help or support:

Professionals did not experience help or support to address actual problems. They were instead referred to simplistic statements that all output would need to be verified and validated before use.

4. Conclusions

What appears to be a significantly expanding use of GenAI by professionals is interesting but not surprising per se. One of the main, human characteristics that make us effective and efficient is our ability to choose NOT to use our intellect to address every problematic phenomenon we encounter; to do so would be extremely wasteful and, potentially, would make us so ineffective that we would be unable to adapt to our environment. Therefore, we apply our intellect and intelligence *selectively,* and even go out of our way to avoid using our thinking and problem-solving capabilities unless we make a judgement that this is necessary. This is, of course, a contributory reason why AI solutions are so popular. The idea of GenAI, and the abilities for which it is promoted, are very seductive to our human experience of life. Thus, we may be motivated by 'wishful thinking' and to accept such tools more readily than perhaps we should. We may become susceptible to disqualifying critical mindsets and ignoring our own misgivings. Having been used to placing reliance on previous generations of productivity tools, such as libraries of reusable code and knowledge based systems, it is not surprising if GenAI appears attractive to professionals.

Ironically, this point is also overgeneralized, and therefore a flawed foundation upon which most GenAI support tools are based. They are fundamentally unreliable and misleading, based on a belief that most people prioritize efficiency over relevance to objective and context, or indeed effectivity. The reason GenAI technologies are not just unreliable but also largely misleading, is the model upon which their functionality is based - generating patterns based on analysis of patterns. There is no actual correlation between training data and output within any specific factual content. This is also the reason why so many people fail, in practice, to do the validation required (if they are honest). The required resources, in time and effort, are not options realistically contextualized enough to be meaningful within the space of a real-world job. If the output generated looks the same as the expected output, then there is no apparent reason, in context, to deviate from habitual prejudice and bias. This will render it emotionally, as well as intellectually difficult to discipline oneself to undertake conscious validation of output with original source, systematically and with the utmost care required. To do this manually demands not just a lot of attention to detail, but a great deal of precious time.

When considering the lack of correlation between any reality and the output from GenAI, it is sobering to reflect that the issues discussed above may become more problematic over time if we continue to abrogate responsibility to check and modify the results. Plausible, but incorrect content will be output into the world and will become a source of new training data for tools of the future. The logic of this is that, eventually, a point of total model collapse would be reached, leaving those whose work has become dependent upon GenAI in an unenviable position.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools in the writing of this work.

References

Alenezi, M. & Akour, M. (2025). Innovations in Software Engineering: A Review of Current Practices and Future Directions. Applied Sciences, 15(3), 1344. https://doi.org/10.3390/app15031344.

Ayinde v London Borough of Haringey (2025). Neutral Citation Number: [2025] EWHC 1383 (Admin), *Report of Case Nos: AC-2024-LON-003062 and CL-2024-000435 in the High Court, King's Bench Division, Divisional Court, 23 May 2025.* Accessed 15 June 2025 at https://www.judiciary.uk/wp-content/uploads/2025/06/Ayinde-v-London-Borough-of-Haringey-and-Al-Haroun-v-Qatar-National-Bank.pdf.

Bahrini, A., Khamoshifar, M., Abbasimehr, H., Riggs, R.J., Esmaeli, M., Mastali Majdabadkohne, R. (2023). ChatGPT: Applications, Opportunities, and Threats, *2023 Systems and Information Engineering Design Symposium (SIEDS),* Charlottesville, VA, USA, pp. 274-279, doi: 10.1109/SIEDS58326.2023.10137850.

Batarseh, F.A., Mohod, R., Kumar, A. & Bul, J. (2020). The Application of Artificial Intelligence in Software Engineering – A Review Challenging Conventional Wisdom. Chapter 10 in *Data Democracy at the Nexus of Artificial Intelligence, Software Engineering and Knowledge Engineering*, F.A. Batarseh & R. Yang (editors). Elsevier.

Bednar, P., Imrie, P., & Welch, C. (2014). Personalized support with ‘little’data. In *Proceedings of Creating Value for All Through IT: IFIP WG 8.6 International Conference on Transfer and Diffusion of IT*, TDIT 2014, Aalborg, Denmark, June 2-4, 2014. pp. 355-358. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-43459-8_2

Ehrndal, M. (2025). How machine learning can dramatically reduce financial instiutions' cost of compliance. BearingPoint. Retrieved 2 May 2025 at https://www.bearingpoint.com/en-se/insights-events/machine-learning-is-the-key-to-efficient-and-effective-aml/

Graham, B. (2024). Forbes Development Council. *Unlocking The Future: How generative AI Is Revolutionizing Software Engineering.* https://www.forbes.com/councils/forbesbusinessdevelopmentcouncil/2024/04/17/unlocking-the-future-how-generative-ai-is-revolutionizing-software-engineering/, accessed 14 April 2025.

Harkess, J. (2025). *AI's Legal Lies Exposed: Combatting the Hallucination Epidemic.* 12 June 2025. Accessed 15 June 2025 at https://www.linkedin.com/pluse/ais-legl-lies-exposedcom-batting-hallucination-epidemic-harkess-v812c.

Hinton, G.E., Osindero, S. & Teh, Y-W. (2006). A fast learning algorithm for deep belief nets. Neural Comput, 18(7),1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527. Jaiswal, C. (2025). What is vibe coding? A computer scientist explains what it means to have AI write computer code – and what risks that can entail. *The Conversation*, 4 June 2025. Accessed 11 June 2025 at https://theconversation.com/what-is-vibe-a-computerscientist-explains-what-it-means-to-have-ai-write-computer-code-and-what-risks-that-can-entail-257172.

Khan, T. (2025). *Law, Lies, and Language Models: Responding to AI Hallucinations in UK Jurisprudence.* The Barristers Group, 12 June 2025. Accessed 15 June 2025 at https://thebarristergroup.co.uk/blog/responding-to-ai-hallucinations-in-uk-jurisprudence.

Kumar, S.V. (2025). Next Generation Software Engineering: A Study on AI-Augmented Development, DevSecOps Frameworks. *Journal of Advances in Computational Intelligence Theory*, 7(2), 5-10.

Morrone M. (2024). *When AI-produced code goes bad.* https://www.axios.com/2024/06/13/genai-code-mistakes-copilot-gemini-chatgpt Nissen, H-E. (2002). Challenging Traditions of Inquiry in Software Practice, in Y. Dittrich, C. Floyd and R. Klischewski (editors) Social Thinking – Software Practice, Cambridge Mass: MIT Press, pp 69-90.

Singh, P. (2024). Software Development with Generative Artificial Intelligence. Scholarly *Research Journal for Interdisciplinary Studies*, 13/84, 106-113.

Surden H. (2024). ChatGPT, Large Language Models, and Law, 92 Fordham L. Rev. 1941 (2024). Available at: https://ir.lawnet.fordham.edu/flr/vol92/iss5/9.

Taylor, J. (2025). Fake cases, judges' headaches and new limits: Australian courts grapple with lawyers using AI. The Guardian, 9 February 2025. https://www.theguardian.com/law/2025/feb/10/fake-cases-judges-headaches-and-new-limits-australian-courts-grappling-with-lawyers-using-ai-ntwnfb?CMP=Share_iOSApp_Other

Yetiştiren B., Özsoy I., Ayerdem M. and Tüzün E. (2023). Evaluating the code quality of ai-assisted code generation tools: An empirical study on GitHub copilot, amazon codewhisperer, and ChatGPT. arXiv preprint: https://arxiv.org/pdf/2304.10778