# Industry leaders reveal how AI is reshaping software development with strategic agility and security focus



Industry leaders in financial services are showcasing how artificial intelligence (AI) is revolutionizing software development, driving both efficiency and innovation while highlighting the importance of cultural and procedural shifts to fully realise AI's benefits. At a recent conference hosted by technology specialist Harness in London, executives outlined five strategic ways their firms are maximising AI’s impact, underscoring the evolving role of developers in an AI-augmented landscape.

One key strategy is fostering flexibility within clear guidelines. At Allianz Global Investors, AI technical lead Dill Bath described using the Open Policy Agent (OPA) engine to codify policies that act as a "copilot" for developers—nudging rather than blocking them towards compliance. This tech-first approach anticipates regulatory changes and aims for agile delivery without compromising standards. Bath emphasised the cultural shift towards platform engineering and granting developers autonomy while maintaining security and audit requirements.

Communication is equally critical in large enterprises. Tony Phillips of Lloyds Banking Group explained the bank's Platform 3.0 initiative, which modernises infrastructure to enable broader AI adoption beyond coding assistance. Philips admitted managing change across thousands of developers is challenging, stressing the need to "hammer home the changes" so that scepticism transforms into belief through tangible successes. Learning from hands-on experience and iterative feedback is vital to integrating AI effectively.

Driving innovation within risk-managed environments is a focus at Hargreaves Lansdown. Senior software engineering manager Bettina Topali highlighted automation's role in embedding guardrails—automated testing, security scanning, and code coverage—that enable faster innovation safely. She urged digital leaders to move beyond buzzwords and visibly demonstrate AI’s value to shift organisational mindsets and keep pace with emerging fintech competitors.

Providing regular feedback to developers about AI-generated code quality is another essential element. Daniel Terry at SEB described how his team equips developers with tools like GitHub Copilot while preparing them for agentic AI, where humans oversee AI agents generating large volumes of code. Terry cautioned novices against "vibe coding," where blind reliance on AI can introduce errors, stressing the importance of testing and governance to ensure secure, compliant software development.

Finally, enterprises must "fight fire with fire" by empowering IT and security teams with AI tools to counter increasingly sophisticated cyber threats. Aaron Gallimore of Global Payments emphasised scalable, secure platforms that reduce developers' overhead in tooling transitions and help audit and security professionals keep pace with AI-driven development. He described educational initiatives aimed at sparking widespread AI adoption and cultivating a culture of ongoing learning.

These practitioner insights align with broader industry data signalling AI’s transformative potential but also its limitations and risks. Surveys indicate nearly 90% of developers regularly use AI tools, mainly for routine coding tasks, which frees them to focus on problem-solving and oversight. AI has been shown to increase productivity and code quality for many, yet trust in AI remains tentative, with less than a quarter of developers strongly confident in AI outputs. Many still prefer peer review and worry about the significant time lost debugging AI-generated code.

Further complicating the picture, recent research reveals that experienced developers working on familiar codebases may actually slow down when using AI tools, as they spend considerable effort reviewing and correcting AI suggestions. However, such findings might not apply to junior developers or new projects, where AI’s support can be more impactful.

Security vulnerabilities in AI-generated code present a critical challenge. Independent studies find nearly half of AI-produced code contains exploitable security flaws, often due to insufficient specification of security requirements during code generation. This risk is exacerbated by "vibe coding," a practice increasingly common but fraught with danger if not properly managed. Experts urge integrating security checks directly into AI workflows, leveraging AI-powered remediation tools, and training developers in secure coding practices to mitigate these risks.

Despite these challenges, AI is reshaping nearly every phase of software development. Automation now extends from coding and refactoring to code review, testing, and debugging—enhancing efficiency, improving error detection, and enabling developers to focus on higher-level creative and problem-solving tasks. Industry commentators advocate establishing new frameworks to ensure responsible AI use that balances innovation with ethical standards and security.

Moreover, certain sectors stand to benefit enormously. The Indian IT industry, for example, anticipates productivity improvements of up to 45% attributable to generative AI over the next five years. Software development roles, in particular, are projected to see productivity boosts around 60%, underlining AI’s strong potential despite ongoing challenges.

In summary, AI’s integration into software development is unmistakably profound, accelerating productivity and innovation while demanding cultural change, strong governance, and a renewed emphasis on security and trust. The future for developers is increasingly collaborative, with AI acting less as a replacement and more as an enhancer—amplifying human expertise, automating routine tasks, and prompting organisations to evolve rapidly to keep pace with technological advances.

### 📌 Reference Map:

* Paragraph 1 – [[1]](https://www.zdnet.com/article/5-ways-you-can-maximize-ais-big-impact-in-software-development/)
* Paragraph 2 – [[1]](https://www.zdnet.com/article/5-ways-you-can-maximize-ais-big-impact-in-software-development/)
* Paragraph 3 – [[1]](https://www.zdnet.com/article/5-ways-you-can-maximize-ais-big-impact-in-software-development/)
* Paragraph 4 – [[1]](https://www.zdnet.com/article/5-ways-you-can-maximize-ais-big-impact-in-software-development/), [[2]](https://www.techradar.com/pro/no-ai-overload-just-yet-googles-new-survey-reveals-how-developers-are-really-using-ai-at-work), [[5]](https://www.itpro.com/software/development/developers-arent-quite-ready-to-place-their-trust-in-ai-nearly-half-say-they-dont-trust-the-accuracy-of-outputs-and-end-up-wasting-time-debugging-code)
* Paragraph 5 – [[3]](https://www.reuters.com/business/ai-slows-down-some-experienced-software-developers-study-finds-2025-07-10/), [[5]](https://www.itpro.com/software/development/developers-arent-quite-ready-to-place-their-trust-in-ai-nearly-half-say-they-dont-trust-the-accuracy-of-outputs-and-end-up-wasting-time-debugging-code)
* Paragraph 6 – [[6]](https://www.techradar.com/pro/nearly-half-of-all-code-generated-by-ai-found-to-contain-security-flaws-even-big-llms-affected)
* Paragraph 7 – [[7]](https://www.forbes.com/councils/forbestechcouncil/2025/04/04/the-future-of-code-how-ai-is-transforming-software-development/), [[4]](https://www.reuters.com/technology/artificial-intelligence/genai-boost-indias-it-industrys-productivity-by-45-ey-india-survey-shows-2025-02-10/)
* Paragraph 8 – [[2]](https://www.techradar.com/pro/no-ai-overload-just-yet-googles-new-survey-reveals-how-developers-are-really-using-ai-at-work), [[5]](https://www.itpro.com/software/development/developers-arent-quite-ready-to-place-their-trust-in-ai-nearly-half-say-they-dont-trust-the-accuracy-of-outputs-and-end-up-wasting-time-debugging-code), [[3]](https://www.reuters.com/business/ai-slows-down-some-experienced-software-developers-study-finds-2025-07-10/)
* Paragraph 9 – [[1]](https://www.zdnet.com/article/5-ways-you-can-maximize-ais-big-impact-in-software-development/), [[4]](https://www.reuters.com/technology/artificial-intelligence/genai-boost-indias-it-industrys-productivity-by-45-ey-india-survey-shows-2025-02-10/)

Source: [Noah Wire Services](https://www.noahwire.com)

## Bibliography

1. <https://www.zdnet.com/article/5-ways-you-can-maximize-ais-big-impact-in-software-development/> - Please view link - unable to able to access data
2. <https://www.techradar.com/pro/no-ai-overload-just-yet-googles-new-survey-reveals-how-developers-are-really-using-ai-at-work> - A recent Google survey highlights the rapid adoption of AI among software developers, revealing that 90% now use AI tools regularly, a dramatic rise from just 14% in 2024. Additionally, 65% of developers report heavy reliance on AI, dedicating a median of two hours per day to such tools. The benefits include increased productivity for 80% of users and improved code quality for 59%. AI is mostly used to handle routine coding tasks, allowing developers to focus more on problem-solving and oversight. However, trust in AI remains cautious, with only 24% of developers expressing strong confidence in AI-generated outputs. The survey underscores that while AI is increasingly integrated, developers still value human expertise and view AI as a support tool rather than a replacement. Google's Ryan J. Salva emphasized that AI acts as both a "mirror and a multiplier," enhancing efficiency in strong teams but exposing flaws in weaker ones. The findings suggest that organizations need to adapt their culture and systems to thrive in the evolving landscape of software development.
3. <https://www.reuters.com/business/ai-slows-down-some-experienced-software-developers-study-finds-2025-07-10/> - A new study by the AI research nonprofit METR found that using AI coding assistants like Cursor actually slowed down experienced software developers when working on codebases they were already familiar with. Despite initially believing AI would speed up their work by 24%, developers took 19% longer to complete tasks with AI assistance. The researchers found that although AI suggestions were often directionally correct, developers had to spend time reviewing and correcting them, which led to the slowdown. This challenges the widespread assumption that AI universally boosts productivity among software engineers—a belief that has spurred significant investment in AI development tools. Previous studies have shown large productivity gains in other contexts, but METR’s findings suggest these gains do not extend to expert developers working on known code. Nonetheless, both study participants and authors continue using Cursor, noting that AI makes development more enjoyable and less mentally taxing, much like editing versus creating from scratch. The authors also noted the slowdown might not apply to junior developers or unfamiliar codebases. This nuance is important as AI continues to reshape the future of software development and the tech workforce.
4. <https://www.reuters.com/technology/artificial-intelligence/genai-boost-indias-it-industrys-productivity-by-45-ey-india-survey-shows-2025-02-10/> - The use of generative artificial intelligence (GenAI) in India's $254-billion IT industry is projected to enhance productivity by 43%-45% over the next five years, according to an EY India survey. The productivity gains will be driven by the internal integration of GenAI within IT companies and the increased adoption of AI in client projects beyond the proof of concept stage. Major companies like Tata Consultancy Services and Infosys have reported that clients are increasingly using AI for new projects, with 89% trialling GenAI and 33% already in production. The survey highlighted that software development roles would see the most significant productivity boost of around 60%, followed by improvements in BPO services (52%) and IT consulting (47%). These three sectors will contribute to 50%-60% of the overall productivity gain in tech services. Executives noted that AI is enhancing customer service, cutting costs, and boosting revenue growth in the industry.
5. <https://www.itpro.com/software/development/developers-arent-quite-ready-to-place-their-trust-in-ai-nearly-half-say-they-dont-trust-the-accuracy-of-outputs-and-end-up-wasting-time-debugging-code> - A recent Stack Overflow Developer Survey reveals that 84% of software developers now use or plan to use AI tools in their daily work—up from 76% the previous year. Popular tools include OpenAI's GPT models, Anthropic's Claude Sonnet, and Google’s Gemini Flash. Despite growing adoption, trust in AI-generated outputs has declined: 46% of developers say they don't trust AI's accuracy (up from 31% in 2024). Many still prefer consulting colleagues, citing the desire to fully understand their code and ethical or security concerns. Debugging AI-generated code is another major issue, with 45% of developers reporting time lost due to buggy code. A Cloudsmith study further highlights that a significant number of developers aren't reviewing AI-generated code before deployment. While Agentic AI is gaining industry attention, only 31% of developers use it, with 17% planning to adopt it. However, 69% of users report increased productivity and efficiency. Despite fears about automation, 64% of developers don’t view AI as a threat to their jobs. Stack Overflow’s CEO emphasized the importance of trustworthy, human-led oversight in AI development, stressing the need for responsible use of curated data to support the evolving role of developers.
6. <https://www.techradar.com/pro/nearly-half-of-all-code-generated-by-ai-found-to-contain-security-flaws-even-big-llms-affected> - A recent study by Veracode found that approximately 45% of AI-generated code contains security flaws, despite often appearing production-ready. The research analyzed over 100 large language models (LLMs) across 80 coding tasks and revealed no significant improvement in security performance among newer or larger models. Java was the most affected language, with over 70% of generated code exhibiting vulnerabilities, while Python, C#, and JavaScript showed flaw rates between 38% and 45%. Common issues included poor defense against cross-site scripting (86%) and log injection (88%). The study highlights a growing risk as AI-generated code becomes more prevalent, with companies like Google and Microsoft reportedly using AI for up to a third of their code. Veracode CTO Jens Wessling emphasized that "vibe coding," where developers rely on AI without specifying security needs, has reshaped software development. The report warns that AI can also help attackers exploit vulnerabilities at scale. To address these issues, Veracode recommends integrating security checks in AI workflows, using AI-powered remediation tools, training developers on secure practices, and deploying firewalls. Wessling stressed that security must be built into AI-driven development processes to avoid accumulating long-term security risks.
7. <https://www.forbes.com/councils/forbestechcouncil/2025/04/04/the-future-of-code-how-ai-is-transforming-software-development/> - The article discusses the transformative impact of AI on software development, highlighting several key areas: Automation of Coding Tasks, Enhancements in Code Quality, Code Generation, Code Refactoring, Code Reviewing, and Testing and Debugging. AI tools are automating repetitive tasks such as code generation, test executions, and debugging, allowing developers to focus on more critical aspects of the software development life cycle. This leads to increased productivity and innovation. AI also assists in analyzing errors, finding vulnerabilities, and improving code quality by identifying issues before the code reaches production. In code generation, AI tools can rapidly uncover potential issues, recommend enhancements, and deliver valuable insights, ensuring high-quality code throughout the development life cycle. AI aids in refactoring by suggesting or automatically implementing changes that improve code clarity, readability, and maintainability. It also provides insights into the security implications and vulnerabilities of code, helping ensure safety and adherence to industry standards. In code reviewing, AI automates parts of the process by suggesting changes and identifying potential mistakes, saving developers time and allowing them to focus on writing high-quality code. AI tools also significantly impact testing and debugging by automating test case generation, creating boilerplate test code, and analyzing test results, thus reducing the overall time required to identify and fix issues. The article concludes by emphasizing the need to evaluate existing practices and establish new frameworks to prioritize responsible AI as part of the development life cycle, ensuring that innovation thrives without compromising ethical principles.