

embarking on the code odyssey:

unveiling the power of AI-assisted programming

Leow Wee Sheng, Yap E Fang, Yong Yam Koon, Vito Chin,
Valerie Chan with contributions from Brett Spedding (Microsoft).

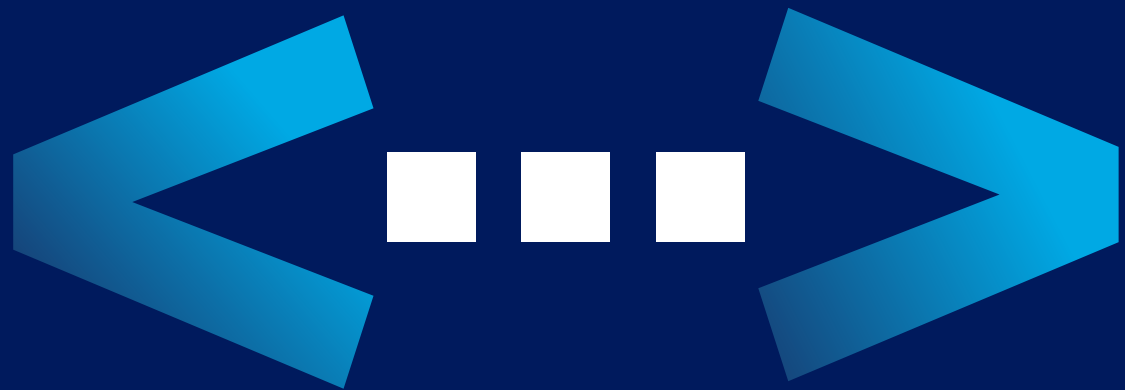





table of contents

AI is a Strategic Imperative	3
Accelerating Software Development	4
Quicker Time-to-Market	5
Quality Improvement	5
Cost Reduction	5
Adopting DevSecOps Methodology	6
Enhanced Code Security	6
Adherence to Best Practices	6
Reduced Technical Debt	7
Empowering Developers for the Future	8
Upgraded Developer Skillsets	8
Higher Employee Retention	8
Reduced Recruitment Pressure	9



1 Embarking on the Code Odyssey: Unveiling the Power of AI-Assisted Programming

AI is a Strategic Imperative

In the ever-evolving landscape of technology, where innovation drives the competitive edge, the role of software development has never been more pivotal. Today, we find ourselves at the intersection of ambition and execution, where harnessing the potential of software innovation is a strategic imperative. The question we now face is not whether to embrace AI, but how to wield it as a catalyst for organisational growth. In this white paper, we shall embark on a journey into the realm of AI-assisted programming and how it revolutionises software development.

The demand for innovation and efficiency in software development has never been greater. Where most industries depend on software as the linchpin of their operations, AI-assisted programming stands as the harbinger of transformation. At its heart, it involves the integration of advanced AI, epitomised by tools like **GitHub Copilot**, **Amazon Q** and **NCS CodeNav**, into the software development lifecycle.

In the pages that follow, we will explore the ways in which the triple A benefits of **AI – Assist, Augment, Automate** – can manifest both horizontally and vertically.

We will delve into how we can usher in a new era of software development through three key value drivers, namely development acceleration, code protection, and developer empowerment. You will gain insights into how AI-assisted programming is crucial in helping companies to maintain a competitive edge.

2 Accelerating software development

AI-powered development tools like GitHub Copilot accelerate coding, improve quality, and empower developers to focus on innovation, driving faster time to market.

Three key value drivers

Traditional software development workflows frequently suffer from inefficiencies and bottlenecks, which have a negative impact on project timeframes and results.

These workflows are riddled with repetitive coding tasks that drain valuable time, debugging processes that typically catch errors late in the game, and lengthy development cycles that can stifle adaptability to changing market conditions.

This section highlights the pressing need for innovation in the software development landscape and sets the stage for examining how AI-assisted programming offers a game-changing remedy to these age-old problems, ushering in an era of unparalleled productivity.

Benefits of AI-assisted programming

- **Faster Coding:** Suggests full functions and code lines, saving time for creative problem-solving.
- **Better Code:** Promotes best practices and catches errors, leading to cleaner and more maintainable code.



Quicker time-to-market

AI-assisted programming follows the shift in preference towards Agile Methodologies – working quickly to make small iterations and get ongoing feedback. Through real-time code suggestions and auto-completion features, these tools simplify the coding process, drastically reducing the need for manual searching and typing. They also assist in the generation of code snippets, the automation of routine coding activities, and the construction of boilerplate code. As such, companies can speed up the software development lifecycle and enable quicker time-to-market in stages such as **Initiation** (*code refactoring*), **Requirement & Design** (*user stories generation*), **Construction** (*code generation*), and **Testing** (*automated unit test scripts*).

Quality improvement

This automation not only accelerates coding but also improves code quality by minimising syntax errors and ensuring best practices are followed. For example, in the case of automated unit test script generation – AI tools like **GitHub Copilot** and **Amazon Q** can analyse the code being developed and provide code suggestions for test cases. It identifies different scenarios and edge cases that should be tested, generating test cases for various input values, corner cases, and boundary conditions. By proactively detecting errors, vulnerabilities, and bugs in real-time, developers reduce the need for debugging efforts later in the development cycle.



GitHub Copilot

Cost reduction

In a **GitHub Copilot** trial conducted to evaluate the impact of AI on developer productivity, it was revealed that participants with access to the AI pair programmer completed the given task **55.8% faster** than the control group¹. Naturally, the productivity gain would imply a significant amount of cost savings – cost savings in project management that arises from shorter development cycles, and cost reductions associated with bug fixing and maintenance, thereby allowing companies to enjoy significant cost reductions.

55.8%
faster
than control
group

3

Adopting DevSecOps methodology

Securing the digital future

The dependence on software applications has reached unprecedented heights as technology is increasingly incorporated into every aspect of our life. As a result, the effects of software flaws, hacks, and poor-quality code have increased tremendously. These problems not only pose serious risks to the confidentiality and integrity of data but also have far-reaching repercussions for an organisation's standing in the public eye, financial security, and legal standing. To navigate this complex and high-stakes terrain, organisations should start recognising the potential of AI-assisted programming in fundamentally altering how code quality and security are maintained throughout the software development lifecycle. In this section, we will delve into the transformative role of AI in ensuring robust code quality and security, safeguarding against potential vulnerabilities, and fortifying the software that underpins our digital future.

Enhanced code security

AI-assisted programming tools like **GitHub Copilot** offer several advantages that go beyond streamlining the software development lifecycle for agile teams. They also contribute to enhancing code security through proactive vulnerability detection and secure code generation. For example, **GitHub Copilot** employs advanced static code analysis and machine learning algorithms to continuously examine the

code being written. It proactively identifies potential vulnerabilities, security loopholes, and coding practices that can expose systems to security hazards. By detecting issues in real-time, it reduces the risk of security and data breaches that can damage the company's reputation and result in legal liabilities.

Adherence to best practices

Tools like **GitHub Copilot** can guide developers to adhere to security coding standards and industry best practices. It offers suggestions for secure coding techniques and guidelines for secure authentication, authorisation, and data protection, helping developers write more secure code from the outset. For example, it can detect and recommend encryption techniques for handling sensitive data in compliance with data protection laws like GDPR or suggest secure authentication mechanisms that align with industry standards. Automated checks for compliance with industry standards and regulations help the company avoid costly penalties and legal complications.

Reduced technical debt

Promoting clean, well-structured code and automated adherence to coding standards help development teams minimise the accumulation of technical debt over time. They prevent shortcuts, hacks, and workarounds that frequently lead to a complicated web of code, making future updates and maintenance increasingly costly and time-consuming. Conversely, AI-driven suggestions guarantee that code is created in a comprehensive and consistent manner, thereby lessening the cost burden brought on by extensive refactoring, codebase rewrites, and extended development schedules, and hence reducing the danger of technical debt accumulation.

GitHub Co-Pilot study
<https://arxiv.org/pdf/2302.06590.pdf>



4 Empowering developers for the future

Empowering developers with AI to innovate and inspire

No longer confined to the traditional realms of coding and debugging, developers are emerging as creative problem solvers and innovators, driving the very essence of technological advancement. In this section, we delineate how AI-assisted programming tools like **GitHub Copilot** have accelerated and empowered this change in developers' function, and how these technologies not only speed up development chores but also serve as invaluable mentors, offering on-the-job learning experiences and exposing developers to a breadth of coding styles and techniques.

Upgraded developer skillsets

AI-assisted programming tools like **GitHub Copilot** foster on-the-job learning opportunities for developers. These tools, powered by machine learning and code analysis, actively assist developers in real-time, suggesting code snippets and offering insights into best practices and emerging technologies. With recommendations spanning a wide range of programming languages, libraries, and frameworks, developers can diversify their skill set effortlessly. By promoting continuous learning within the development workflow, developers not only gain proficiency in current technologies but also remain primed to embrace emerging trends. This translates to a more innovative and agile organisation, ultimately contributing to the company's competitive edge and long-term success in the ever-evolving digital landscape.

Higher employee retention

Employees who appreciate professional development may find these tools that assist continual learning and skill improvement appealing. In fact, 71% of workers surveyed in a Gallup study felt that job training and development increased their job satisfaction, and 61% indicated that upskilling opportunities are an important reason to stay at their job. Hence, the use of such tools may in turn result in a higher retention rate. What further establishes the long-term retention of employees is the fact that these tools reduce the monotony commonly associated with repetitive and time-consuming coding tasks. Empowering

developers of the future to automate routine tasks and instead focus on more meaningful and intellectually stimulating aspects of their work increases job satisfaction. For instance, a survey conducted revealed that 60-75% developers who utilised **GitHub Copilot** felt more fulfilled with their job, felt less frustrated when coding and were able to focus on more satisfying work. Similarly, 73% of developers reported that the tool helped them stay in the flow, and 87% of them could preserve mental effort during repetitive tasks.¹

Reduced recruitment pressure

Overall, offering cutting-edge tools can make a company more attractive to top tech talents, as skilled developers are more likely to stay with a company that provides them with the tools and resources they need to excel in their roles. The ability to attract and retain these talents is crucial today as programmer scarcity is an increasingly pressing concern for organisations, with the global tech talent shortage expected to reach 85.2 million by 2030². Developer shortage has stalled new projects in recent years, but AI-assisted programming tools like **GitHub Copilot** offer a compelling solution.

By upskilling existing talent, **Copilot** not only mitigates the recruitment pressure but also elevates the company's agility and cost-efficiency. Developers are provided with instant access to a wealth of coding knowledge, gaining assistance to tasks ranging from routine to complex. The acceleration of development cycles also enables current team members to handle more tasks with greater efficiency.

This, in turn, reduces the need for extensive external hiring and helps organisations to navigate the developer shortage with confidence.



¹ From: <https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>

² <https://www.griddynamics.com/global-team-blog/software-developer-shortage-us>

nCS  **make
extraordinary
happen**

nCS.co