

GenAI Service Offerings



Data & AI
Azure



Digital & App Innovation
Azure



What you should know about Generative AI for software delivery



GenAI tooling can help write code faster, and improve the developer experience.

But the mileage varies based on the circumstances, and experience still matters to uphold stability and quality at the same time.



GenAI tooling can boost the whole delivery cycle, not just coding.

Coding is only one element of the overall software delivery process, and it is not the one that takes the most time. Further opportunities to reduce waste and friction with GenAI exist across the end-to-end lifecycle from feature definition to deployment.



Good engineering practices matter more than ever.

Engineering discipline not only helps mitigate the quality risks of generated code, but is also crucial to deal with the increased throughput. It is also important because GenAI tools amplify what you already have - it can make the good better, and the bad even worse.



Thoughtworks can help you get started.

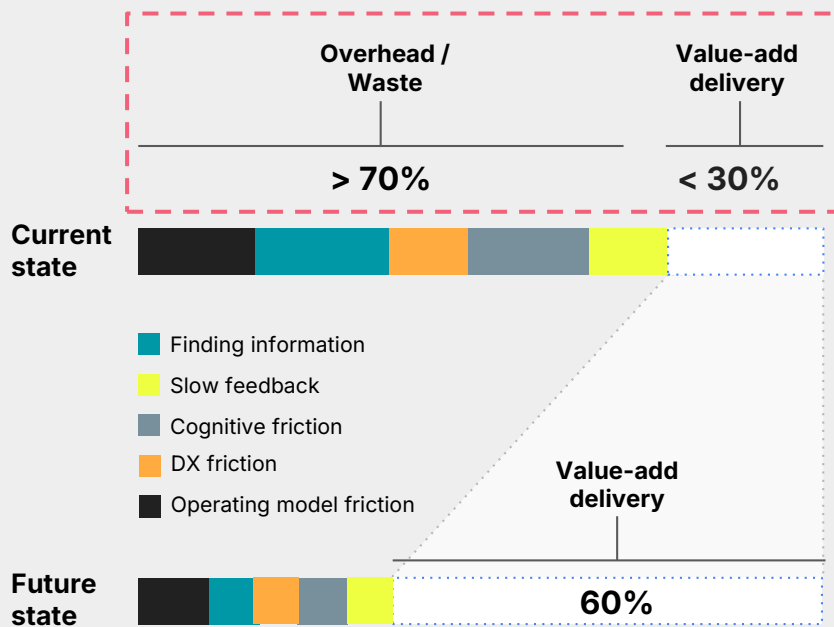
We can share our learnings and assets with you. We can help to establish a change lead; establish guidelines and policies to manage risk; establish measures and mechanisms to collect learnings and experiences; run a pilot

Gen AI can remove impediments to flow

The most value to unlock lies in the areas of waste and overhead in the delivery process.

Probable GenAI Impact	Examples of common friction that are draining productivity and block flow
HIGH	Finding information - Improved freshness and clarity of documentation and searchability across enterprise
MED	Developer Experience friction - acceleration of code and test generation, more context available directly in the toolchain
MED	Slow quality feedback loops - support for test generation, and brainstorming gaps of specifications and tests earlier; complex and brittle architectures will still be hard to test
LOW	Cognitive overload / task switching - some gains from tools, but also trade-offs with the constant need to review and understand inputs from GenAI tools
LOW	Operating model friction - better structured requirements will help, but inter-team dependencies and ways of working for flow cannot be impacted with GenAI much

Team effectiveness



Engineering practices still matter, if not more.



Good practices help deal with bottlenecks from higher coding throughput.



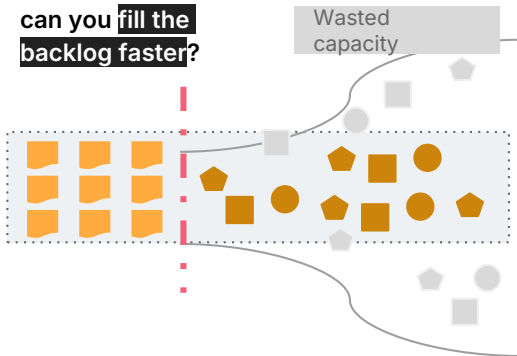
GenAI amplifies your status quo - the good gets even better, but the bad can get worse.



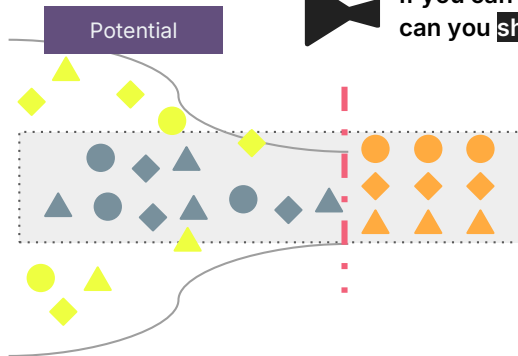
Good practices mitigate the GenAI risks, and help manage the quality of more code.



If you can code faster, can you fill the backlog faster?



If you can code faster, can you ship faster?



If you can produce more code, can you also keep your technical debt in check?



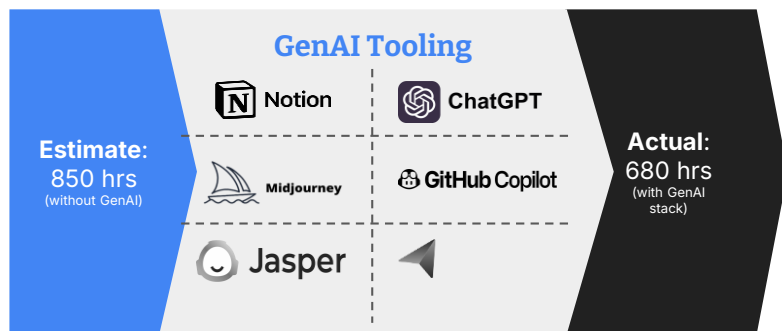
If you can produce more features, how are you tracking their value?

GenAI has clear potential to increase speed and productivity

Developers see the benefits.

StackOverflow	GitHub	Thoughtworks
<p>2023 Developer Survey: "77% feel favorable towards using AI tools as part of their development workflow"</p>	<p>Survey: "70% of developers see a benefit to using AI coding tools at work"</p>	<p>We're using coding assistance at 20 clients and growing; over 600 of our developers have had exposure to the tools.</p>

A Chinese digital giant got to Product-Market-Fit 20% faster



Case study: 20%-30% productivity increase

Thoughtworks internal research experimenting with AI in software delivery

Objective	Assess how using ChatGPT and GitHub Copilot improves productivity across the software delivery lifecycle																											
Preliminary results	<p>20%-30% Overall Productivity Improvement</p> <p><i>Mileage will vary</i></p> <table border="1"> <thead> <tr> <th colspan="3">Analyze</th> <th colspan="3">Architect</th> <th colspan="3">Code</th> </tr> <tr> <th>Feature Design</th> <th>Feature Breakdown</th> <th>AC Definition</th> <th>Architecture Design</th> <th>Domain Modelling</th> <th>API Design</th> <th>Code Design</th> <th>Coding</th> <th>Post-coding</th> </tr> </thead> <tbody> <tr> <td>↑10-30%</td> <td>↑10-30%</td> <td>↑10-30%</td> <td>↑5-10%</td> <td>↑5-10%</td> <td>↑10-30%</td> <td>↑5-10%</td> <td>↑10-30%</td> <td>↑10-30%</td> </tr> </tbody> </table>	Analyze			Architect			Code			Feature Design	Feature Breakdown	AC Definition	Architecture Design	Domain Modelling	API Design	Code Design	Coding	Post-coding	↑10-30%	↑10-30%	↑10-30%	↑5-10%	↑5-10%	↑10-30%	↑5-10%	↑10-30%	↑10-30%
Analyze			Architect			Code																						
Feature Design	Feature Breakdown	AC Definition	Architecture Design	Domain Modelling	API Design	Code Design	Coding	Post-coding																				
↑10-30%	↑10-30%	↑10-30%	↑5-10%	↑5-10%	↑10-30%	↑5-10%	↑10-30%	↑10-30%																				

Limitations
<ul style="list-style-type: none"> - Developer expertise is essential for evaluating the validity and accuracy of inconsistent results. - Prompting expertise is critical for generating high-quality prompts that lead to improved outcomes. - Limitations in first-generation LLM tools can increase cognitive load by requiring developers to maintain context across interactions.

AI-assisted software delivery examples at Thoughtworks

Coding assistance with GitHub Copilot

20+ clients

Over 600 Thoughtworkers have gained experience with GitHub Copilot, including daily work with codebases of over 20 of our clients worldwide.

We've published a "[cheatsheet](#)" version of our **Getting Started Guide**.

Org-specific coding assistant

Payments Provider

Prototyped a tool for coding assistance that uses a self-hosted and fine-tuned open sourced LLM for the client.

Product ideation

Boba AI

A [copilot application for product strategy](#) and generative ideation leveraging ChatGPT.

Story boarding

Our designers are using Midjourney to generate story boards and other user experience design artefacts.

GenAI-assisted QA & Testing

Financial Services Company

Building Generative AI bots to generate diverse and realistic test scenarios, create test data, and generate automation and performance test code using ChatGPT. Goal is to improve testing efficiency and more comprehensively validate the system's capabilities.

Team AI

Innovation incubator product

Teams shape this prompt composition application to help with tasks like user story writing, planning, or generating code using ChatGPT. Makes good prompt engineering practices reusable across the whole team. Helps with knowledge exchange, upskilling, and shorter feedback loops.

Reverse Engineering mainframe code

"LegacyBridge" Proof of Concept

How to make the Reverse Engineering practice in mainframe modernisation more effective - this Proof of Concept using ChatGPT is inspired by work at one of our Financial Services clients, and explores the transformation of COBOL code into documentation and Java code.