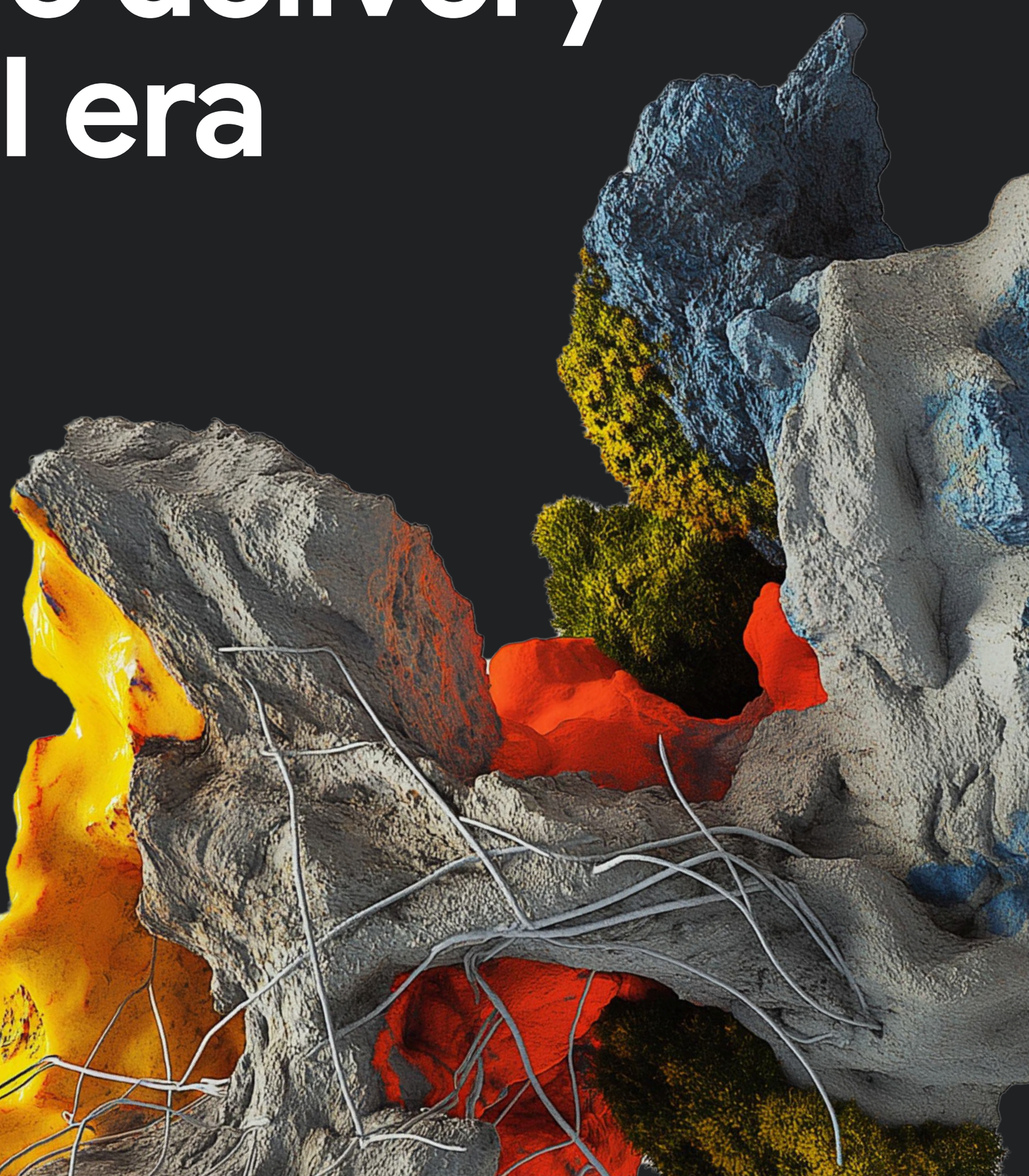
Google Cloud

Beyond code

Your guide to accelerating software delivery in the AI era



It’s 2030.

The majority of code is generated by AI.

Your developers focus on creative and strategic thinking.

Developers delve into solving complex and intricate challenges that require abstract thought.

They dedicate more time to improving the user experience, developing software that’s intuitive and enjoyable to use.

And developers continually improve AI systems to be fairer, more transparent, and even more secure.

The path towards unlocking developer innovation, collaboration, and productivity starts today.

We’ll show you how.

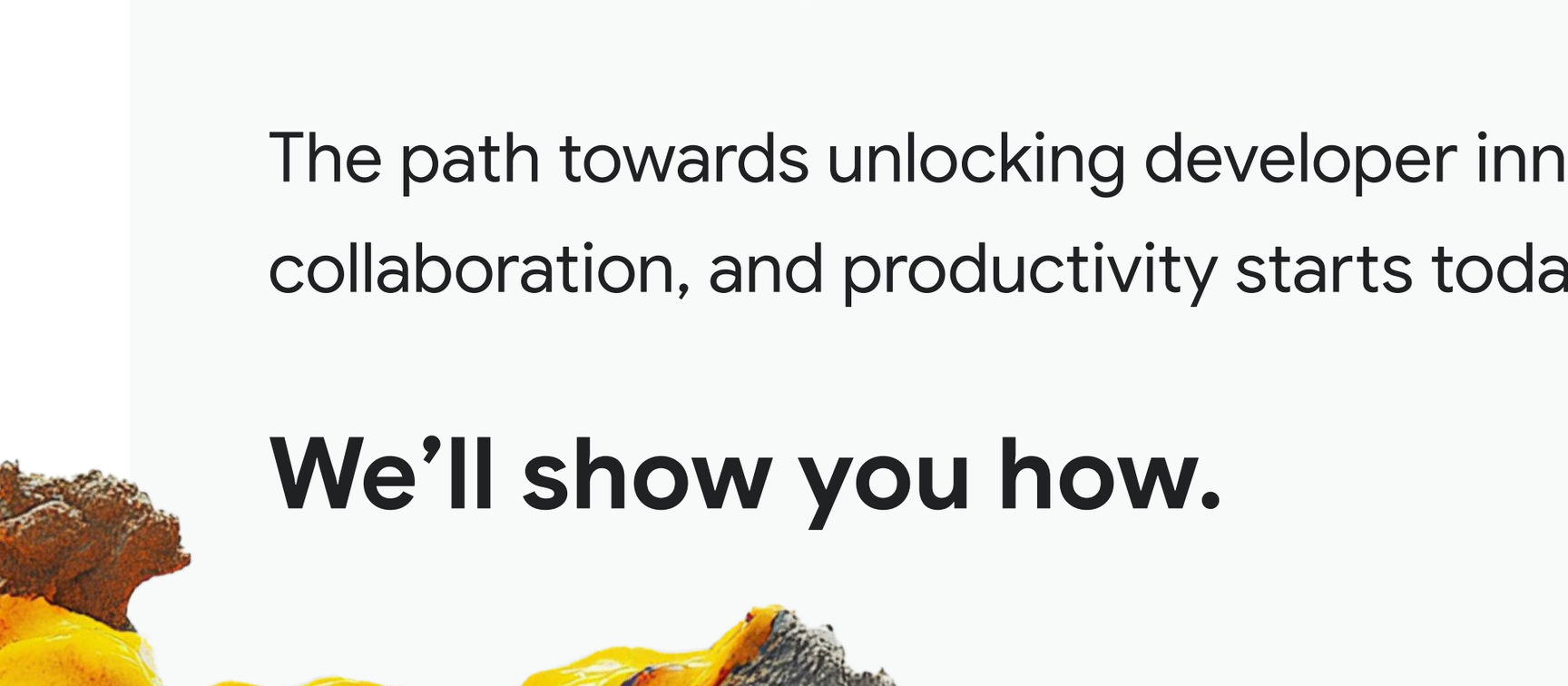


Table of contents

Set the stage for integration success 07

1. Assess organizational readiness 09
2. Start with high-impact areas 10
3. Prepare your pipeline to scale 11
4. Start with pilot programs 12

Harness your best asset—your developers **14**

5. Ease the transition through effective change management 16
6. Adopt agile and DevOps practices 17
7. Promote ongoing education and training 18

Measure. Improve. Repeat. **21**

8. Track developer productivity 22
9. Monitor code accuracy 23
10. Review security regularly 25

Introduction

Welcome to the new era of software development. Today, generative AI-powered tools are helping boost developer productivity by reducing toil, ultimately elevating customer experiences and driving innovation. IDC predicts that by 2027, AI will be capable of automatically generating code to meet functional business requirements for 80% of new digital solutions.[1]

Toil is a major hindrance to efficient software delivery. And especially with the emergence of gen AI, the security, compliance, and access management demands are only increasing. When developers are mired in manual, repetitive, and automatable tasks, they're not creating enduring value for your business.



“If a human operator needs to touch your system during normal operations, you have a bug. The definition of normal changes as your systems grow.”

Carla Geisser
Google Site Reliability Engineering



AI coding assistants are powerful tools for reducing toil within your software teams, but integrating them into your existing teams and processes requires a considered strategy to ensure maximum value for you and your teams.

Succeeding with this new way to work requires developer training and new ways of measuring effectiveness.

> We'll show you how to use gen AI when building modern apps in 10 actionable steps. **So you can get started now. Because your competition isn't waiting.**

Set the stage for integration success





Gen AI has already begun assisting developers —generating code, automating deployment, and optimizing development. But as with any new tool, many developers still have questions about how to best use gen AI to accelerate software delivery safely and securely.

This is where your adoption strategy truly yields dividends. Getting your organization off to the right start will help ensure you can maximize the benefits of AI assistants.

17%

of global organizations have introduced gen AI-enhanced apps[1]

38%

are investing significantly in gen AI[1]

Here are 4 steps you can take to make sure gen AI pays off in your development workflows.

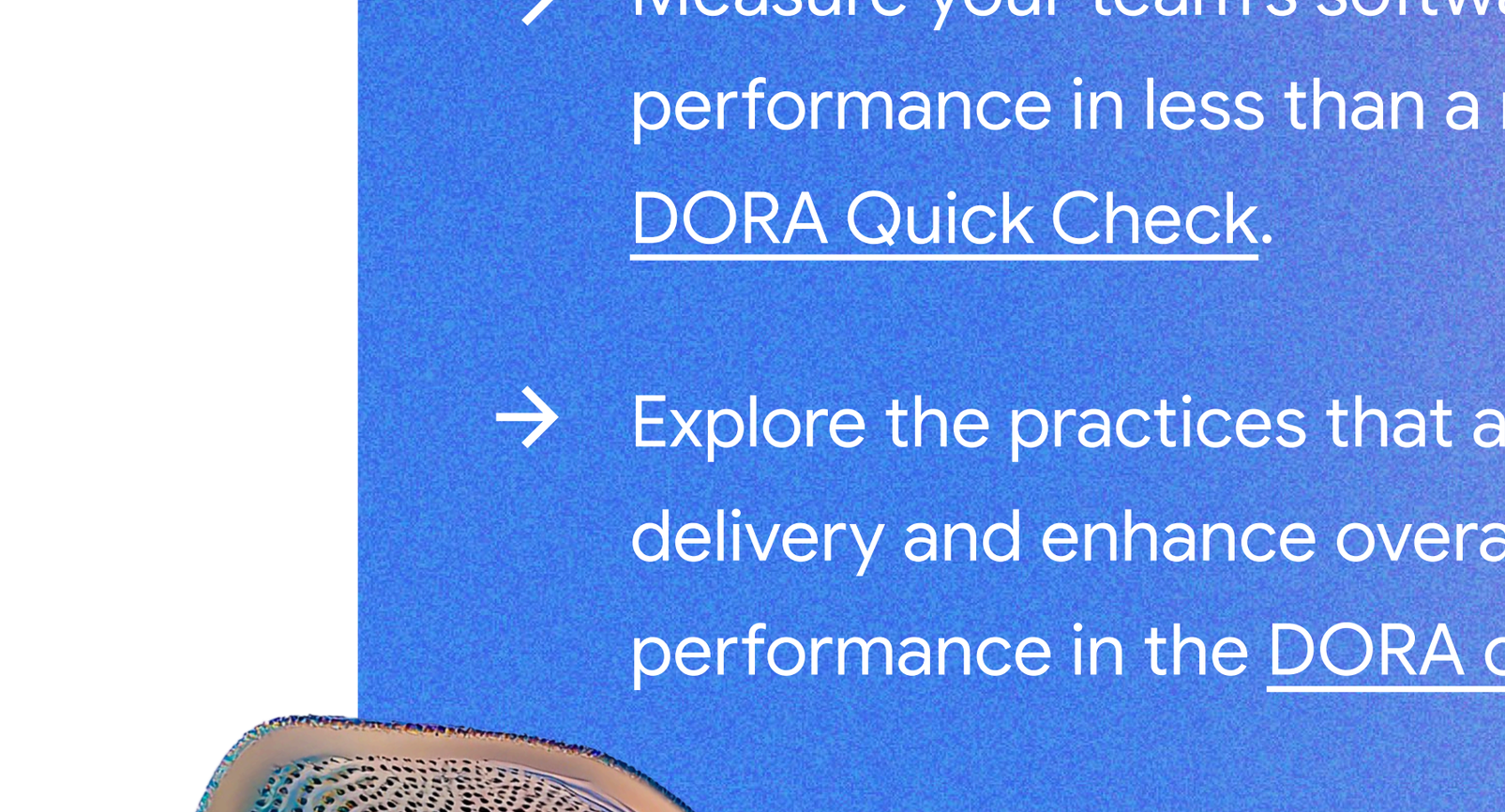
1

Assess organizational readiness.

Start with an understanding of your current state of software delivery processes using methodologies like DevOps research and assessment (DORA).[2] This framework helps you measure your software delivery performance, identify the best areas for improvement, and determine your readiness for AI adoption.

→ Measure your team's software delivery performance in less than a minute with the DORA Quick Check.

→ Explore the practices that accelerate software delivery and enhance overall organizational performance in the DORA capabilities catalog.



2

Start with high-impact areas.

Prioritize quick wins and areas of the business that will see the most impact when considering where to start with AI assistants for software development.

The top three use cases for AI coding assistants are:[1]

51%
Unit test case generation

41%
Code generation

40%
Code autocompletion

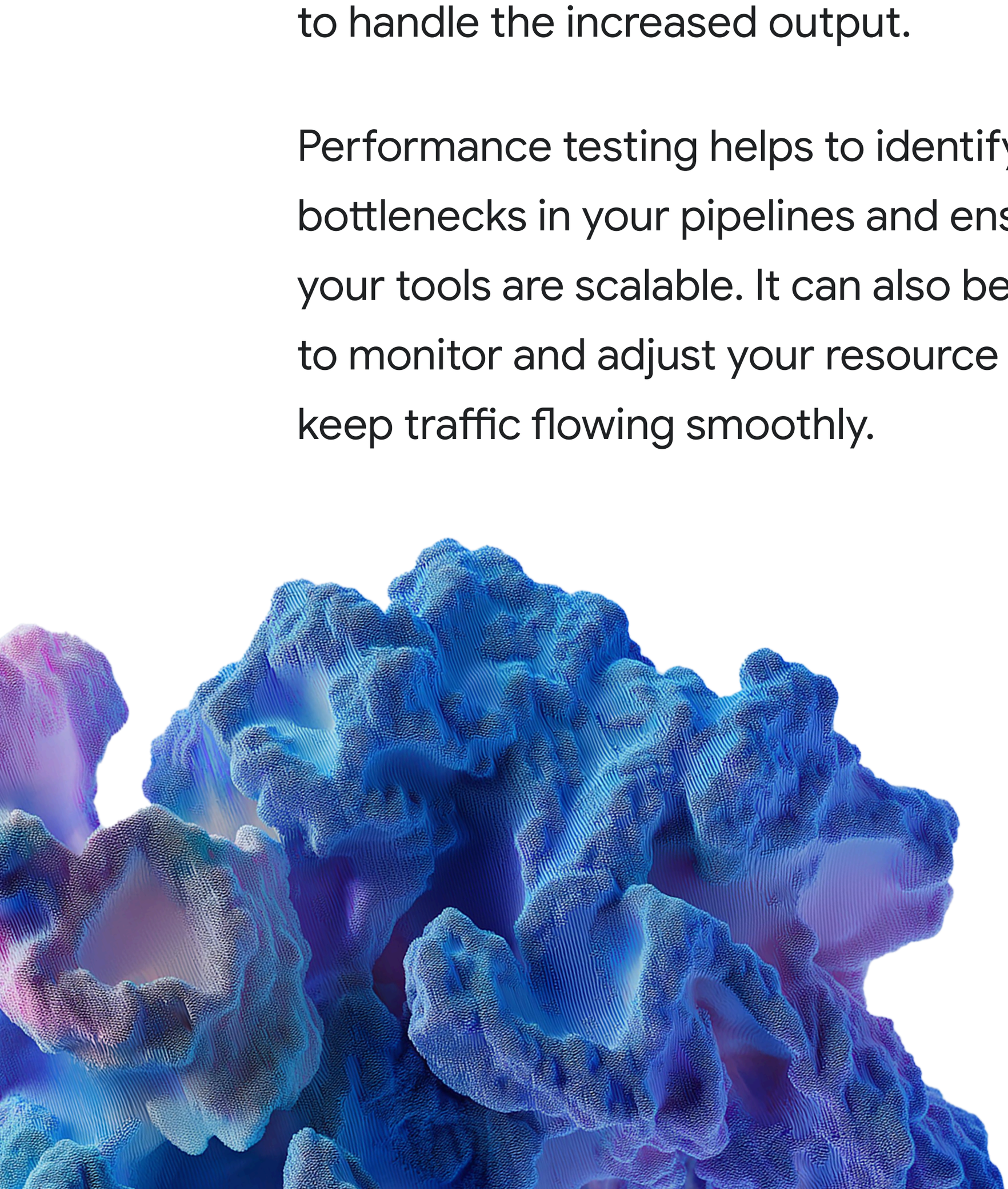


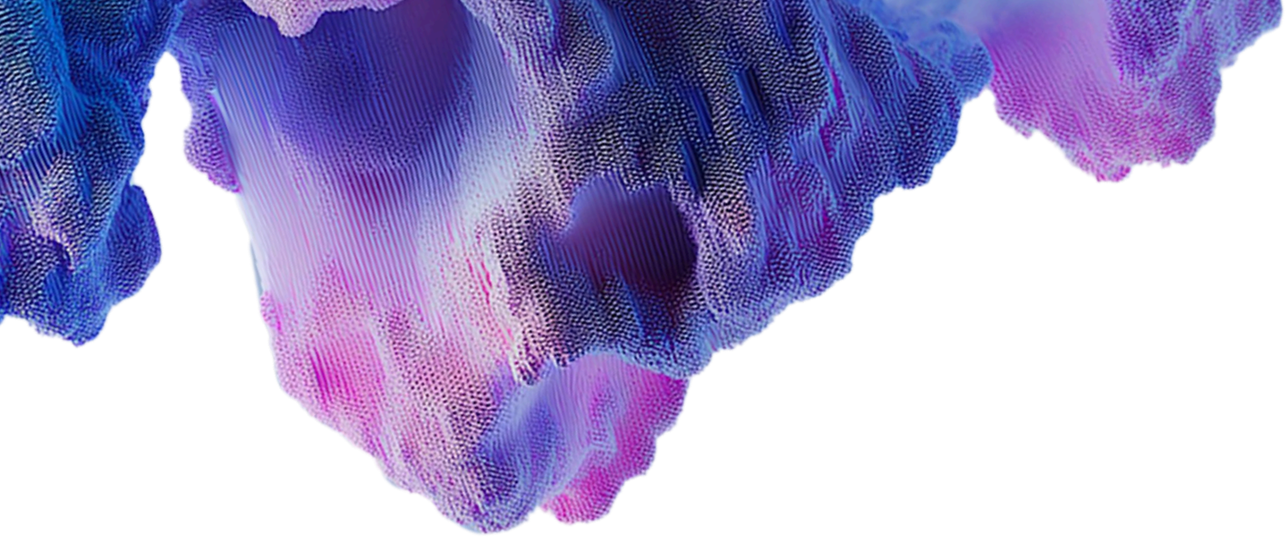
3

Prepare your pipeline to scale.

AI-powered tools are expected to speed up code generation, so it’s important to also make sure your CI/CD pipelines are ready to handle the increased output.

Performance testing helps to identify any bottlenecks in your pipelines and ensures your tools are scalable. It can also be helpful to monitor and adjust your resource usage to keep traffic flowing smoothly.





4

Start with pilot programs.

A pilot program enables you to experiment with AI tools in a controlled environment and helps you to gather feedback and make necessary adjustments before a full-scale rollout. Here are some tips for pilot success.

Create a diverse team.

Your pilot team should be a representative microcosm of your larger development organization, including varying seniority, cross-functional representation and adopters of diverse technology stacks.

Establish key metrics.

The metrics should align with your defined objectives for the AI coding assistant across the development lifecycle, such as: productivity, code quality, developer experience, and business impact.

Continuously monitor impact.

Improvements of AI tools on development processes help you make iterative improvements and use learnings for future integrations.

> By establishing clear baselines before the pilot begins and following the above tips, you can gain a comprehensive understanding of the AI coding assistant's impact on the development lifecycle and make data-driven decisions about its wider adoption.

Start a pilot program with Gemini.

Every use case is unique. And so it’s worth dialing in on the right processes and parameters that will work for your business. Begin with a pilot team and use these learnings to scale AI innovation further—and share knowledge throughout your organization.

Gemini makes it easy to get started.



Turing builds on pilot success.

“The pilot study we did previously with Code Assist in our development teams has revealed very promising results, with productivity boosts of around 33% in pull request merge per developer per week.”

Jonathan Siddharth, CEO & Co-founder, Turing

Watch the full story

Harness your best asset—your developers



Technology teams have the most to gain.

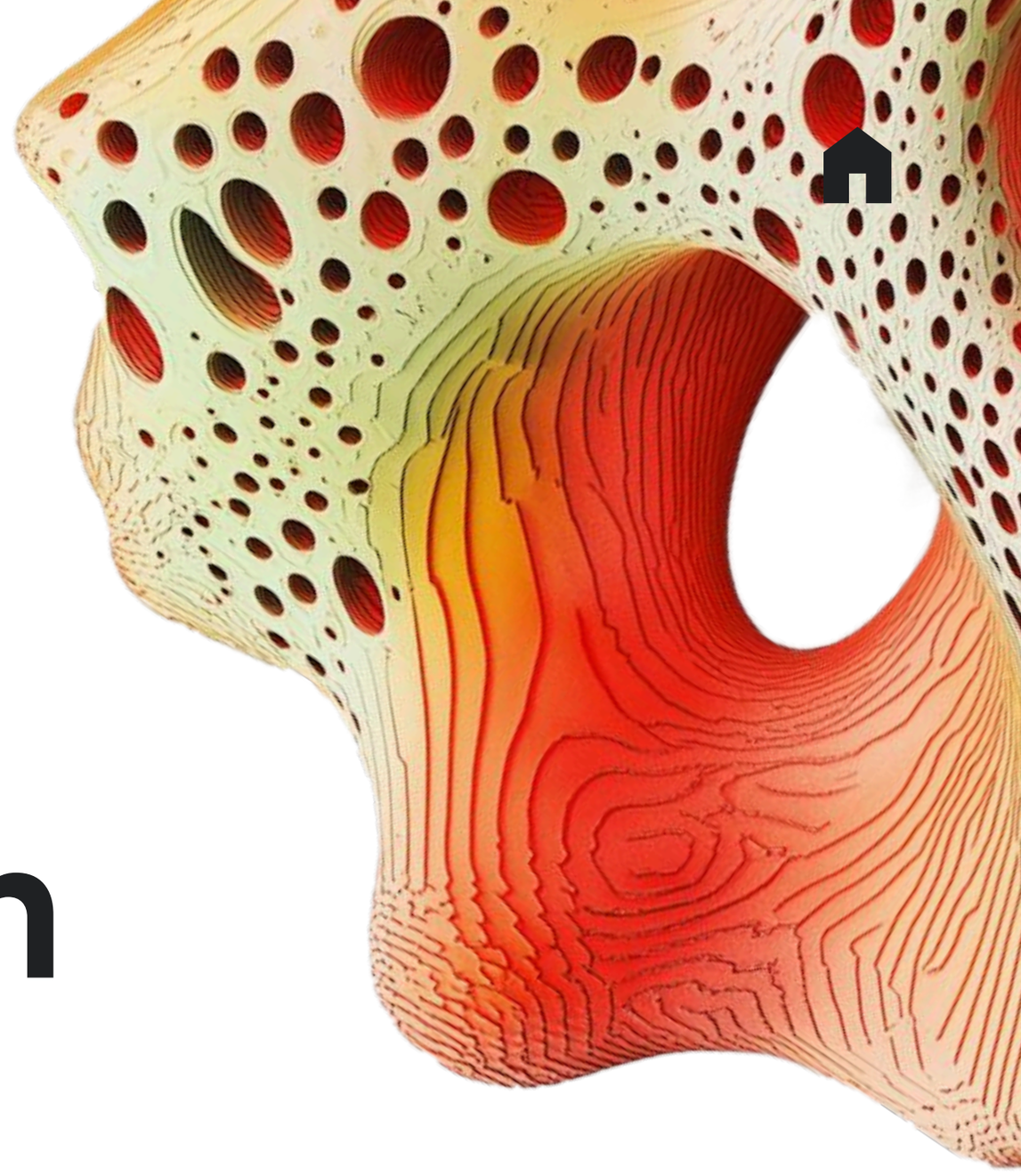
Gen AI has the greatest impact on technology decision-making, beating finance, operations, customer experience, and other business areas.[1]

AI-powered development tools augment human capabilities. They're designed to help developers with the heavy lifting, alleviate repetitive tasks, and bring new products to market quickly to keep pace with market demands.

Gen AI can't replace human ingenuity. When developers are freed from mundane tasks, they can spend more time innovating and problem-solving.

> The best way to instill confidence in your teams during the transition to using gen AI tools is by ensuring they have the right training, support, and knowledge to use the tools effectively.

Here are three key steps to help prepare your development teams.



5

Ease the transition through effective change management.

Gen AI tools enable developers to be more productive and focus on higher-value work.

Top benefits of AI assistants for developers:[1]

49%
Improved collaboration

35%
Increased job satisfaction

34%
Accelerated resolution of development-related issues

While developers will reap these benefits, they may currently have concerns around job security or questions about how their roles may change. Addressing these cultural and psychological aspects of AI adoption can ease the transition.



6

Adopt agile and DevOps practices.

Strengthening agile and DevOps methodologies within your organization promotes seamless AI integration and enhances existing workflows. This includes automating test cases, continuous integration, and continuous development practices. Develop clear guidelines for both the ethical and principle uses of gen AI for development teams. This includes human oversight, which is still a vital loop of the process, as well as model bias checks, and data transparency and traceability.

Start with a scalable approach to experimentation.

Large-scale change is only achieved incrementally. To reap the benefits of gen AI, developers need to learn how to use the tools effectively and share their learnings from pilots and experiments.

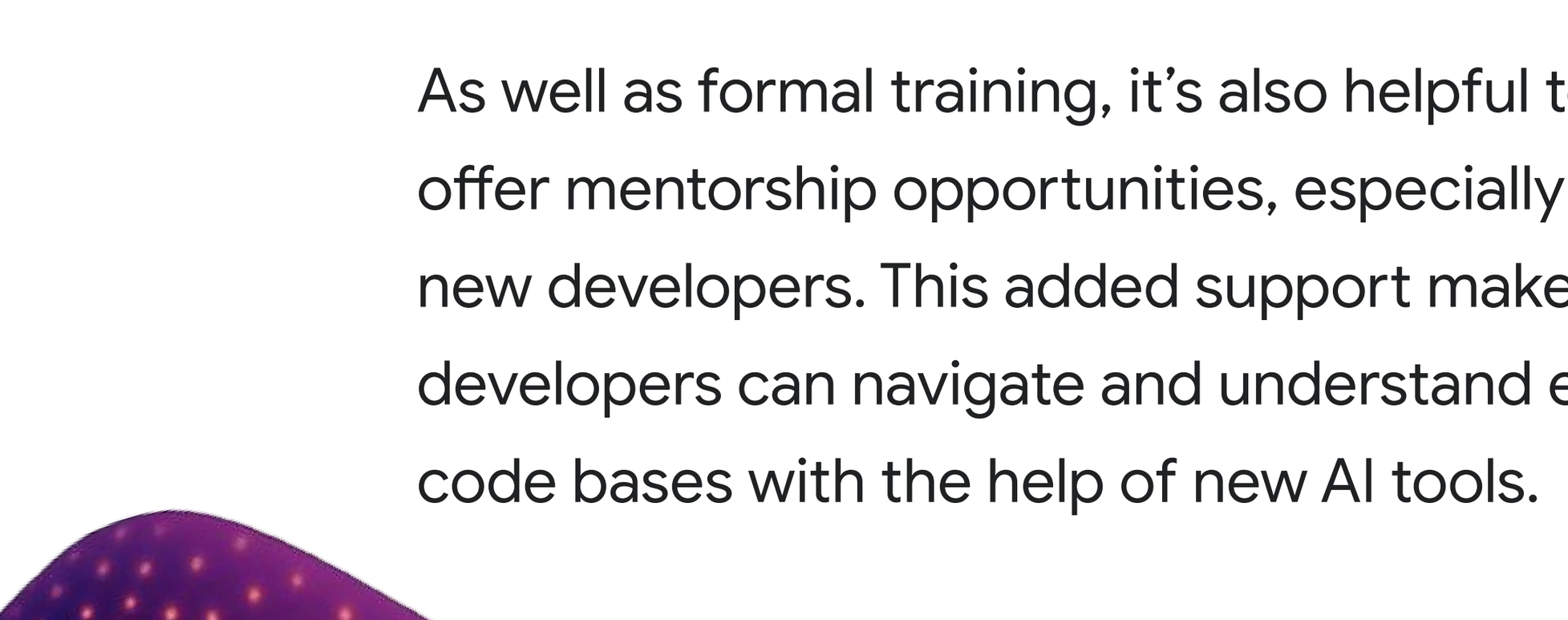
Support your team by building the right community structures to spread knowledge.

7

Promote ongoing education and training.

As AI continues to reinvent how we work, investing in training programs helps developers understand and effectively use the latest tools and best practices. Training should include familiarizing developers with new workflows and best practices for AI to enhance productivity. Give developers permission to tinker, explore, and make incremental progress in a safe and scalable environment.

As well as formal training, it's also helpful to offer mentorship opportunities, especially for new developers. This added support makes sure developers can navigate and understand existing code bases with the help of new AI tools.



Gemini Code Assist boosts teams' productivity.

With Gemini Code Assist, your developers can receive help generating code and optimizing performance, while understanding legacy code faster and more easily. Gemini can help development teams increase speed and productivity, while also bolstering security.

Sign up for our workshop →

Wayfair developers focus on more satisfying work.

"Gemini Code Assist brought in significant improvements across the spectrum. For example, developers were able to set up environments 55% faster than before, there was over 48% increase in unit test coverage for the code, and 60% of developers reported that they were now able to focus on more satisfying work."

Fiona Tan, CTO, Wayfair

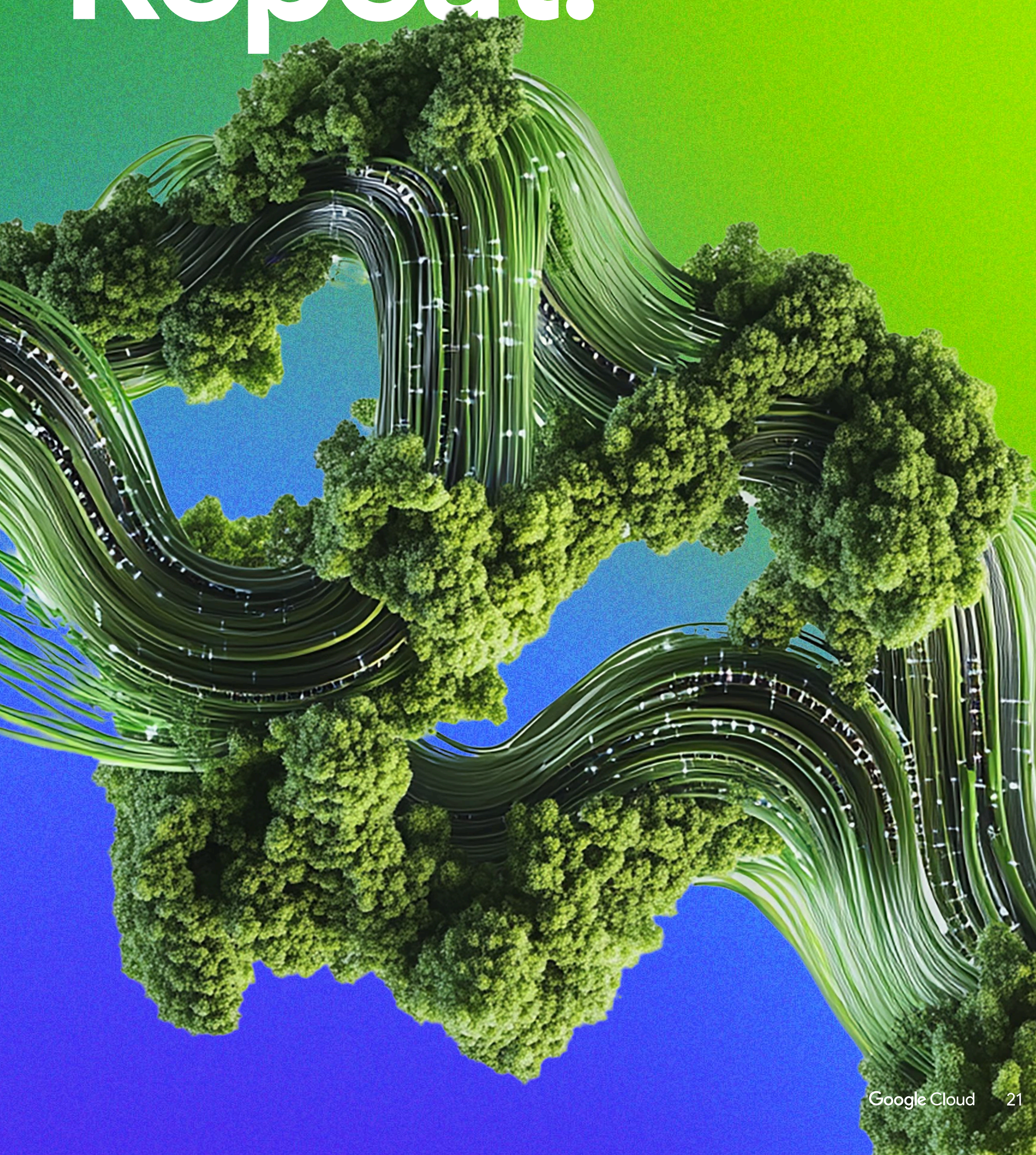
Watch the full story

“We are at the cusp of a paradigm shift in software engineering, itself. It’s going to be beautiful.”

Jonathan Siddharth
CEO & Co-founder, Turing



Measure. Improve. Repeat.



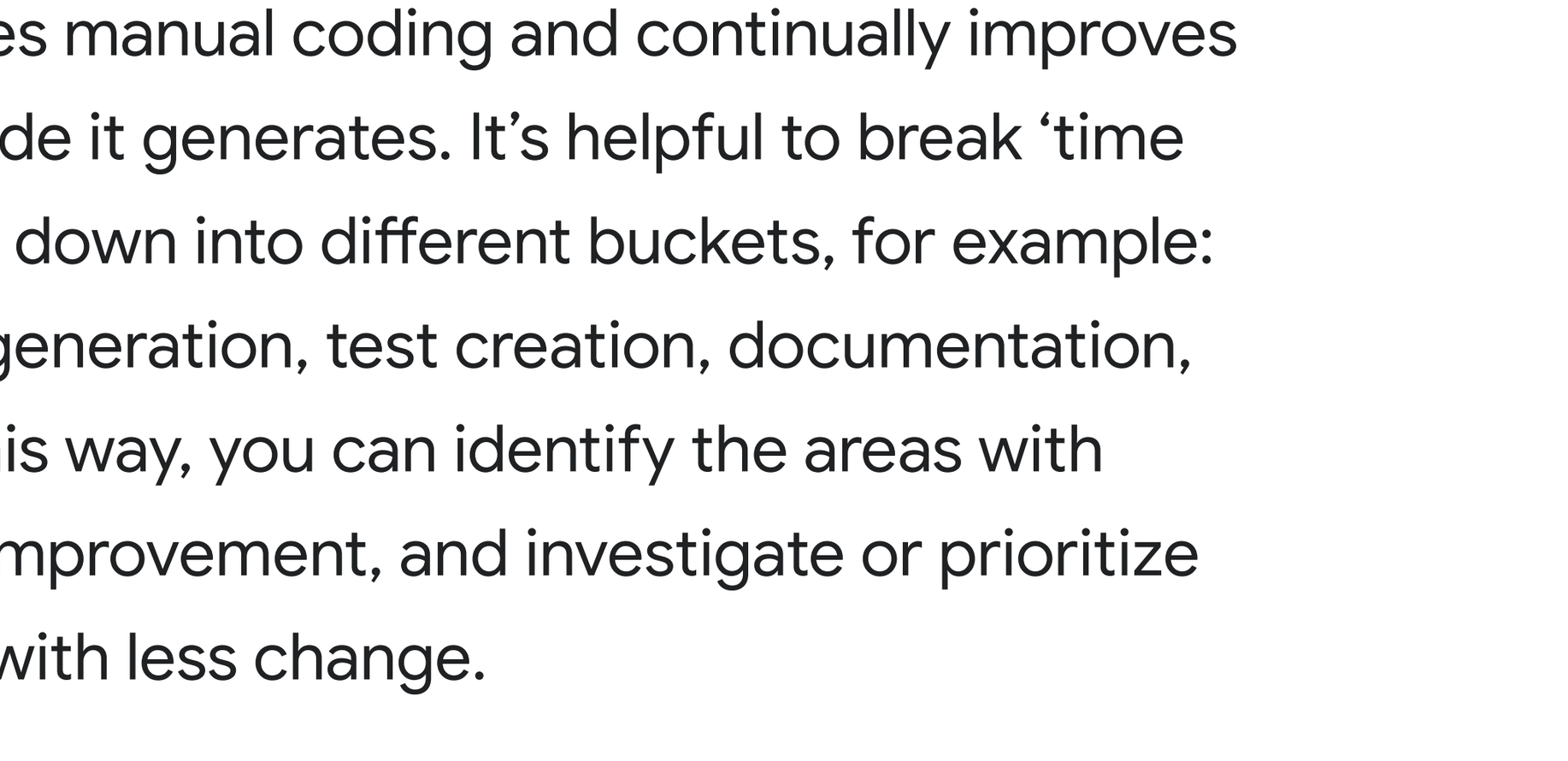
> As you iteratively introduce AI into your workflows, it's important to track both its effectiveness and your progress.

Here are three steps you can take to improve your monitoring.

8

Track developer satisfaction.

One way to gauge productivity gains and developer satisfaction is with regular surveys of those using the new tools. How can you spot gains if you aren't measuring results? **Keep track of common pain points as well as time saved**, so you can tell how effectively gen AI reduces manual coding and continually improves the code it generates. It's helpful to break 'time saved' down into different buckets, for example: code generation, test creation, documentation, etc. This way, you can identify the areas with most improvement, and investigate or prioritize areas with less change.

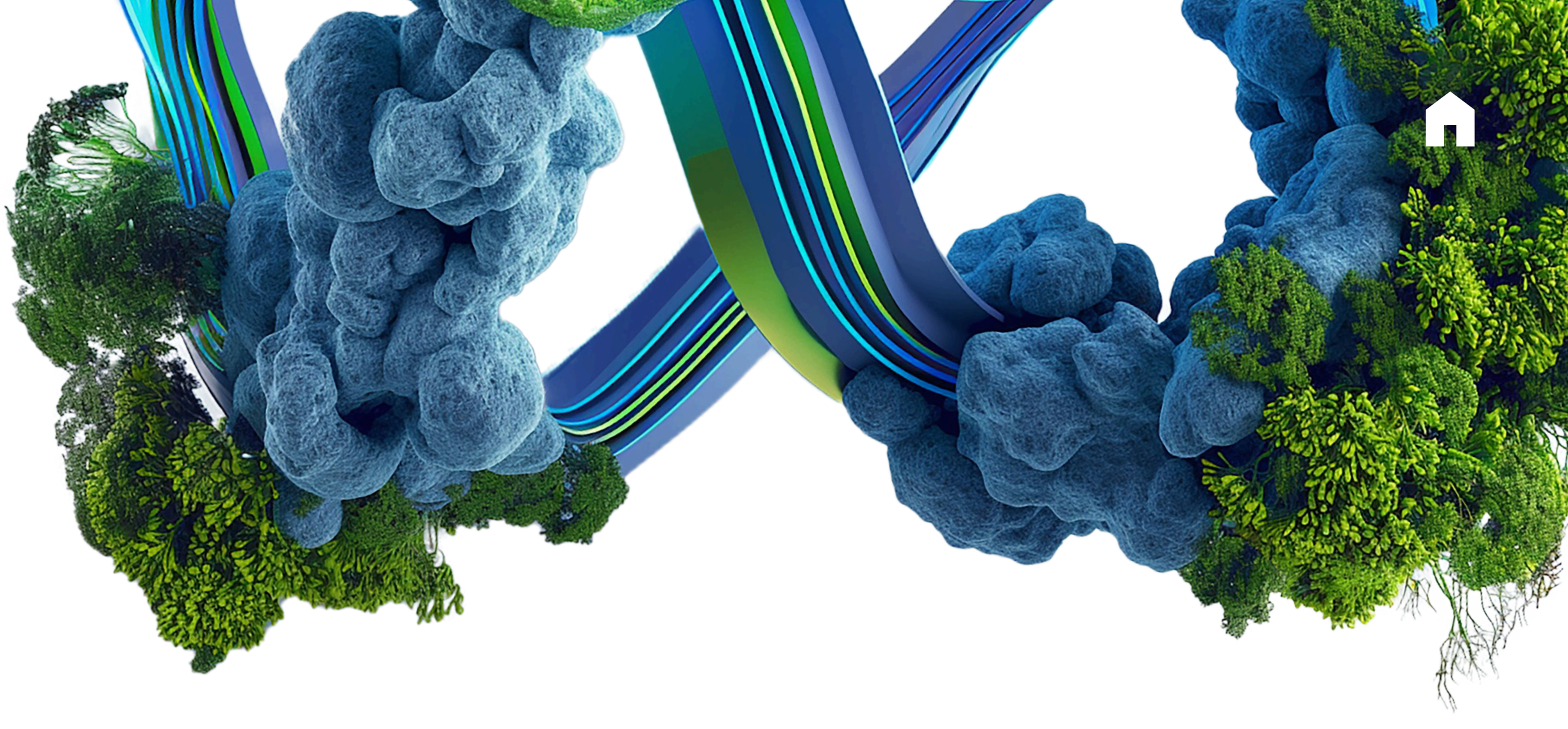


9 Monitor code accuracy.

Autocomplete rates and AI models are improving all the time, and there may come a day when AI-generated code is production-ready. However, the human feedback loop will always be a vital step in ensuring code quality, detecting bugs, and adhering to responsible AI practices.

Track the percentage of generated code that is functional and requires minimal to no modification. You can also **identify the number of bugs**, both before and after gen AI adoption, to shed light on the tool's ability to improve code quality. **Emphasize human reviews** of gen AI-created code to ensure it adheres to coding standards and best practices. And **establish regular bias checks** on models to ensure outputs remain fair and objective.





Build developer trust.

A developer's trust in AI tools is paramount to their confidence in creating applications. Training a model with context from your own codebase reinforces understandable, correct, and trustworthy AI models. You can also explore 'temperature' options to control and customize the output of your AI models. Temperature refers to the degree of randomness in your responses, where lower temperatures are good for prompts that require direct responses, and higher temperatures are better for diverse and creative responses.

Another important factor in establishing trust is gaining buy-in internally and addressing concerns of job security. Consider having an internal rollout plan with messaging that encourages the use of AI in your organization, positioning AI as a developer enhancement and not a replacement.

10

Review security regularly.

Along with maintaining human code reviews, implement security measures such as:

- Input validation to prevent malicious code
- Output monitoring to detect unexpected behavior
- Vulnerability scanning to identify and proactively remediate potential security risks

Staying on top of your security capability will help you stay ahead of emerging threats.

Be led by the latest.

For over a decade, the DORA Accelerate State of DevOps report has been offering critical insights into the practices and capabilities that fuel the success of high-performing technology organizations.

The 2024 report examines the impact of AI, platform engineering, and more on software development and delivery performance.

Download the 2024 DORA report →





Conclusion

AI can be a powerful tool in reducing toil and elevating developer capabilities. Moving from initial decision to business benefit takes careful planning, support, and strategy.

You're holding the practices that can improve your software delivery by leaps and bounds. What will you and your developers create with the time you'll save?

Want to get started? **Get in touch**.

[1] Generative AI in the Software Development Life Cycle: An IT Leader's Guide. (2024) IDC.

[2] How to transform your organization. DORA.

Google Cloud

© 2025 Google LLC 1600 Amphitheatre Parkway, Mountain View, CA 94043