Microsoft Azure

# Accelerating Developer Productivity

## How four companies found the right tools to empower developers and boost productivity

This is for you if:

- ✓ You lead development teams and are tasked with technology and tools decisions for your org.
- ✓ You need to invest in technology that helps developers be more productive and satisfied—while ensuring business goals are met.
- ✓ You want to learn how other companies are solving challenges similar to your own.

# Empower developers to build easier, collaborate better, and deploy faster

Decision makers are already challenged to find the right tools for their teams, but the emergence of new AI-powered developer tools such as GitHub Copilot has added a new sense of urgency. GitHub research predicts that increases in developer productivity due to AI will boost global GDP by over $1.5 trillion by 2030.[1] Organizations of all sizes want to reap the clear financial and productivity benefits that come with adoption of AI-assisted technologies.

As companies race to integrate AI into their already complicated tech stacks, they face a wide range of challenges, and a common theme emerges—**multivendor tool usage**. The rapid evolution of new tech innovations compels many orgs to adapt by stitching tools and projects together, creating new challenges such as:

- Security risks and vulnerabilities.
- Collaboration concerns and siloed workflows.
- Productivity and time losses.
- Tool sprawl, delayed deployments, and increased costs.

In this e-book, we'll look at how four companies used the **Microsoft developer stack** to solve these business challenges, improve operational efficiency, and boost security. You'll learn how these companies:

- Improved developer productivity and experience by taking advantage of AI- assisted development tools.
- Decreased friction and secured app lifecycle management in workflows with natively integrated end-to-end security.
- Created a unified tech ecosystem that easily integrated with their existing tools.
- Found solutions that were easy to roll out and readily adopted by their teams.

The **Microsoft developer stack** brings together some of developers' favorite coding and collaboration tools—GitHub, the Visual Studio family, and Azure—to offer a comprehensive set of development tools that accelerates productivity, elevates performance, and frees developers to focus on what they do best—creating.

[1]Thomas Dohmke, "The Economic Impact of the AI-powered Developer Lifecycle and Lessons from GitHub Copilot," GitHub Blog, June 27, 2023, https://github.blog/2023-06-27-the-economic-impact-of-the-ai-powered-developer-lifecycle-and-lessons-from-github-copilot/.

# Empowering developer creativity with AI

## Industry pioneer General Motors fosters a culture of test and learn

"We want developers to get there right away with all the right tools. Dev Box, GitHub Copilot, and GitHub Advanced Security all play really well together for accomplishing this goal."

**Lakshmi Jagan**

Director of Enterprise Technology Services
Inventory Automation and Development,
GM Information and Digital Technology

Automotive giant GM has demonstrated unique adaptability in their industry, shifting from a legacy auto maker to a pioneer in the field of sustainable energy and the manufacture of electric vehicles (EVs). The company has positioned itself as a global leader in the EV field by adopting a developer-centric, software-first mindset that has spurred them to reimagine their developer stack and onboarding experiences.

GM decided to create agile test-and-learn environments to empower developers. And not just to increase their productivity but also to spark their creativity. The result has been a faster pace of innovation and new capabilities that align with the company's mission and business goals.

To do this, GM used the Microsoft developer stack: a collection of cloud-based services from Microsoft Azure, Visual Studio, and GitHub, including Microsoft Dev Box, Azure Deployment Environments, GitHub Copilot, and GitHub Advanced Security.

# The challenges

### Security risks and vulnerabilities

- The company wanted to get new developers to their first pulls faster without taking security shortcuts.
- Developers needed a higher degree of confidence in the security of their code.
- Third-party developers needed access to internal resources.
- In-house devs needed to be granted security access for each new project.

### Collaboration concerns and siloed workflows

- Teams couldn't easily view, work, or collaborate across all stages of the development lifecycle.
- It wasn't simple to spin up test-and-learn environments in order to validate code before it went into production.

### Productivity and time losses

- New projects weren't able to get up and running quickly, in large part because GM didn't have an all-in-one solution with the tools, processes, and security settings already provisioned, preconfigured, and ready to code.
- It took 2–3 weeks to onboard new developers.
- The process for granting devs access to new projects was lengthy because of security protocols and administrative overhead.

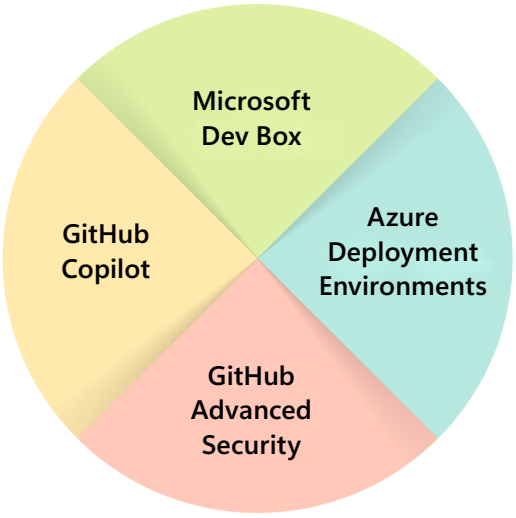
"Nobody expected that it would be that easy for them to develop, deploy, publish—everything. So, they're very happy about it."

**Ravi Sambangi**

Staff Researcher,
GM Advanced Analytics Center of Expertise

# Solutions used

Adopting a software-first mindset using the Microsoft developer stack has been an important part of GM's transformation into an EV leader. In addition to expanding developers' capabilities and productivity with flexible coding, seamless collaboration, and productivity-boosting AI tools, GM was also able to streamline administration and reduce security vulnerabilities in its processes and code.



# Outcomes

### 1. Boosted dev creativity and productivity with real-time security and AI

To provide devs with greater agility and confidence in the stability of their code, GM implemented **GitHub Advanced Security**, a set of automatable security tools that analyzes code in real-time and provides vulnerability alerts. They also used **GitHub Copilot**, an AI tool that provides code suggestions based on the context of the application. This freed GM developers from boilerplate structure and formatting, letting them generate the mundane code more quickly so that they could spend more time on bigger ideas.

### 2. Streamlined developer access while using Zero Trust security principles

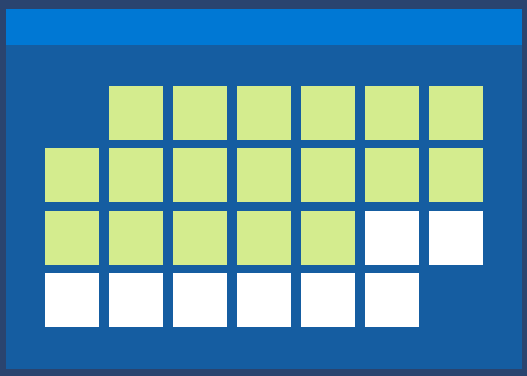
GM used **Microsoft Dev Box** to streamline developers' access to the environments they needed while also maintaining peace of mind for data security. It allowed devs to use different network connections for different Dev Box images, while keeping them safely in bounds by limiting access to sensitive resources. And it allowed admins to split work across two Dev Box images—one with access granted, the other without—and to select the specific images devs should use.

### 3. Improved developer agility and flexibility

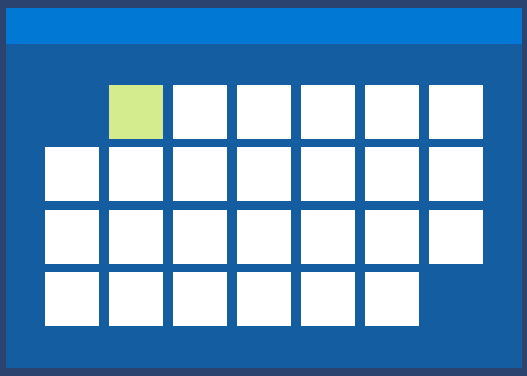
**Dev Box** also provided the flexibility devs require to be the most productive. Each Dev Box can be configured with a different set of tools, allowing devs to spin up a base image and start work immediately. They could extend the capabilities of the base image as needed and add their favorite tools, if company policies allowed it.

## Reduced onboarding time with Microsoft Dev Box

Before adoption: 2-3 weeks



After adoption: 1 day or less



### 4. Accelerated dev productivity with agile test-and-learn environments

**Azure Deployment Environments**, part of the Dev Box portal, also became an integral tool for GM—enabling the use of infrastructure as code with the deployment environment. This allowed dev teams to quickly spin up environments to test code before shipping it.

### 5. Empowered new devs to contribute from their very first day on the job

And, there was one more big win from **Dev Box**: GM was able to reduce developer onboarding time from 2–3 weeks to within one day. Now devs get access to all the tools, capabilities, and endpoints they need to do their job right on day one.

Learn more

# Securing what matters

## Gjensidige—A 200-year-old brand built on trust

"For 200 years, we've helped safeguard the lives, health, and assets of our customers. We need to remain just as vigilant in protecting their data. Trust is essential to our brand—we take it very seriously."

**Rida Aatif**

Head of the Cloud Platform
and Infrastructure Team, Gjensidige

Gjensidige, an insurance company headquartered in Norway, has a rich history dating back to 1816. It's built a legacy of trust safeguarding the assets and data of its customers across Scandinavia. But rather than rest on a storied past, the company has looked to the future and embraced digital transformation—currently offering most of its services exclusively online. This necessitated a renewed commitment to meticulous end-to-end security solutions to keep its customers' data safe.

The company set out to build a new application platform that their developers would want to use—while integrating security into dev projects from the very beginning. Using a combination of Microsoft Azure, Azure Kubernetes Service (AKS), GitHub Advanced Security, and Microsoft Defender for Cloud, they enabled developers to write more secure code, respond more quickly to vulnerabilities, and adopt best practices to secure their Kubernetes environments.

# The challenges

### Security risks and vulnerabilities

- Gjensidige wanted to adopt modern DevSecOps solutions but needed to ensure they integrated well with their existing Kubernetes environment.
- To meet European compliance requirements in the digital era, the company needed to evolve its existing security protocols and break some old habits.

### Collaboration concerns and siloed workflows

- Siloed workflows between app developers and IT security teams resulted in security issues being identified late in the development process.
- Security data was stored in multiple places and wasn't easily accessible to everyone who needed it.

### Productivity and time losses

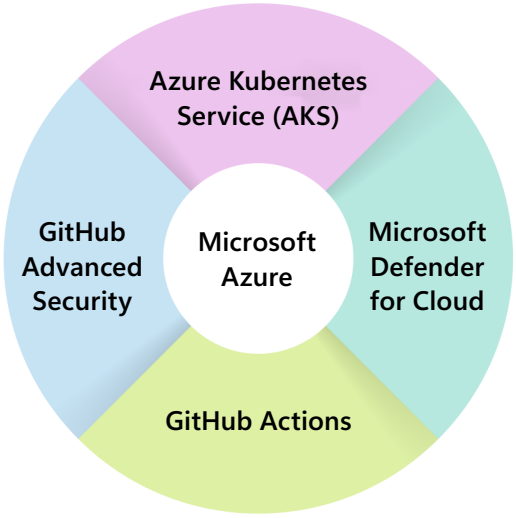
- Security issues were often caught at the end of the development cycle, rather than throughout the process, making it more complex and time-consuming for devs to fix.

### Tool sprawl, delayed deployments, and increased costs

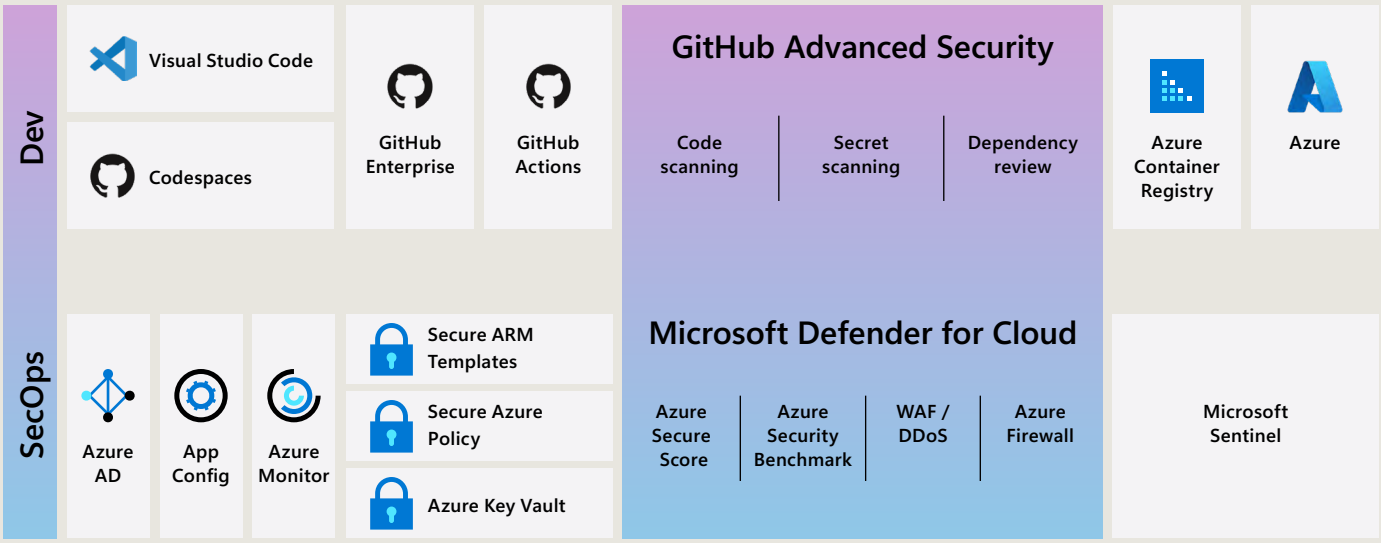
- They needed to ensure that new cloud-based security services would integrate seamlessly with the existing DevOps pipeline and Kubernetes environment—and would also allow for interoperability with open-source and other third-party tools.

# Solutions used

Using a full collection of solution components from Microsoft and GitHub, Gjensidige successfully implemented a modern DevSecOps solution for their application platform—empowering their dev and security teams to collaborate and share security responsibilities.




## DevSecOps solutions integrate security from the beginning of the development cycle



# Outcomes

### 1. Built a unified DevSecOps solution to develop apps securely

Using **GitHub Enterprise** and **Microsoft Azure**, Gjensidige built a modern application platform that incorporates DevSecOps best practices. They took advantage of developer-friendly tools in **GitHub Advanced Security** and **Microsoft Defender for Cloud** that integrated well into existing dev workflows. This created a culture of empowerment where developers were equipped to write more secure code, embrace security best practices, and share responsibility for security.

### 2. Eliminated silos between development and security teams

Gjensidige's breakthrough in cross-team collaboration came when they increased their security teams' visibility into the development pipeline. Using **GitHub** as the single source of the truth and tools integrated with **Microsoft Defender for Cloud**, security teams gained a single, unified view of security—and a common set of security-related metrics and key performance indicators (KPIs) to collaborate with dev teams on. They were able to monitor the outputs of **GitHub Advanced Security** scans, view reports on container security, and pinpoint the teams that needed guidance.

### 3. Embedded security into developer workflows

The company was able to generate enthusiasm and buy-in among developers for the new platform by incorporating security as a continuous part of their workflow. By automating previously manual tasks, such as repo creation, using **GitHub Actions**, Gjensidige made it easy for developers to implement security guardrails with minimal effort. With automatic code scanning, secret scanning, and dependency reviews performed by **GitHub Advanced Security**, developers could rely on consistent, predictable security automation in their workflows, freeing them up to focus on creative work.

"In the past, security issues were often caught at the end of the development process, where they took a lot more time to fix. Today, with automated security controls embedded into their workflows, developers get instant feedback and can fix things right away."

**Rida Aatif**

Head of the Cloud Platform and Infrastructure Team

### 4. Integrated a Zero Trust security approach in a Kubernetes environment

Since they were already using **Azure Kubernetes Service (AKS)**, Gjensidige was able to take advantage of how AKS makes it easier to patch and redeploy containers, as well as eliminates the need for developers to manage credentials.

Learn more

# WOptimizing workflows and fostering collaboration

## Vodafone telecommunications becomes more nimble and less siloed

"Your ability to respond, react, to stay relevant, to innovate, and to compete comes down to your ability to release value to your customers as frequently as possible. That's why Azure DevOps is such an important underpinning for business success and for cultural change. It really has revolutionized how we work."

**Ben Connolly**

Global Head of Cloud Engineering and
Head of Digital Engineering UK, Vodafone

Vodafone global telecommunications company has been on a journey to transform their company with a more agile and efficient operational model. Historically, Vodafone's IT operations were secondary to their network priorities, and were characterized by siloed workflows and infrequent software releases. As Ben Connolly, Global Head of Cloud Engineering notes, "Today, software has become the business. As a result, we knew we had to make changes."

Vodafone set out to make these organizational changes knowing they would need to transform every level of IT, including who they recruited, how they collaborated, and even how they motivated each other. To accomplish this cultural shift to a more agile approach, they started by standardizing their DevOps on Azure and GitHub to drive consistency and improve performance across the business.

# The challenges

### Security risks and vulnerabilities

- There were disconnects between security teams and developers, and they had no automated ways to roll back a deployment, identify lines of code that were flawed, or link a flaw to a specific developer before code was deployed. They could only deploy "up to a point," and then had to hand it back to the security team for penetration testing.

### Collaboration concerns and siloed workflows

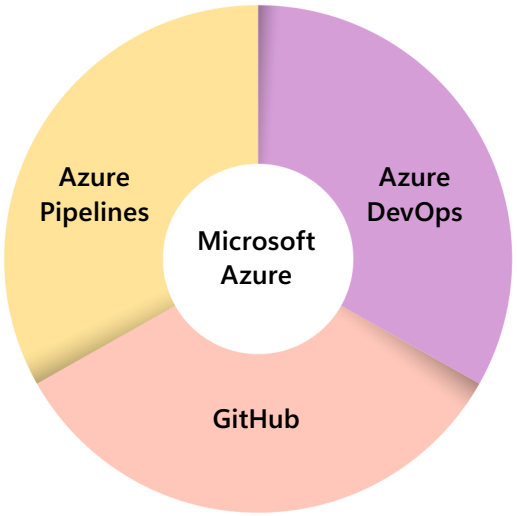
- Different teams were using disparate systems, including different source repositories. They also used change control systems that relied on Word documents. This made cross-team processes excessively time-consuming, discouraged collaboration, and created a risk-averse atmosphere that hindered innovation and agility.

### Productivity and time losses

- Large deployments required 12 to 16 people to be up all night, and then required them to remain on call for days to fix issues as they cropped up.

# Solutions used

With a 50 percent reduction in both engineering time and defects, and a hundredfold increase in software release capacity, the transformation Vodafone has achieved is nothing short of extraordinary. Since implementing GitHub and Azure DevOps services, their once-siloed testing teams are now able to collaborate easily, deliver more value, and deploy much faster using self-service tools, flexible integrations, open-source technologies, and DevOps practices.




"One of our most important assets is engineering hours. With GitHub and Azure DevOps, we are now in a position to streamline and automate environments, coding, and testing so that our engineers focus more of their time on building new customer features."

**Ahmed El Sayed**

UK CIO and Europe Digital Engineering Director, Vodafone

# Outcomes

### 1. Accelerated release frequency and improved release quality

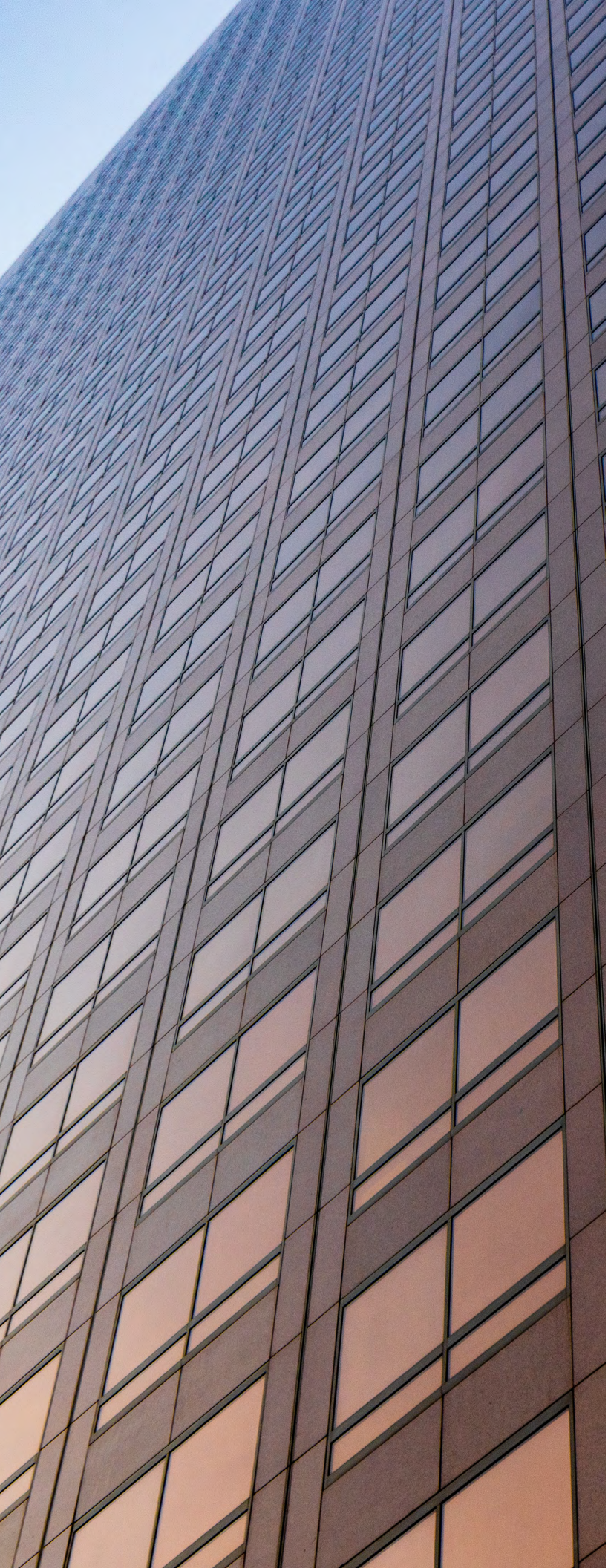
Vodafone's shift to optimized workflows and improved collaboration using **Azure DevOps** tools has transformed the way it releases software changes. Their global engineering team has gone from being able to do five releases per year to 500 releases per month.

### 2. Increased developer productivity without adding headcount

By integrating **GitHub** and **Azure DevOps**—particularly **Azure Pipelines** within Azure DevOps—Vodafone was able to fully automate testing and feedback to engineers. As a result, their rollbacks started to happen automatically. The company estimates its use of GitHub and Azure DevOps has resulted in a fourfold increase in developer productivity, as measured by the rise in the number of releases and the improvement in output quality—without any corresponding increase in their workforce.

### 3. Enhanced developer efficiencies via code sharing

Through automation and global code reuse of more than 200 web components, Vodafone's web engineering team achieved a 40 percent improvement in time savings. The app engineering team achieved a 50 percent improvement in time savings, as well as a 50 percent reduction in code defects and reworks, all made possible by using **GitHub** and **Azure DevOps**.





## 4. Automated security pipelines to move toward DevSecOps practices

Since implementing **Azure Pipelines**, instead of multiple handoffs between deployment and penetration testing, the security teams now own their part of the pipeline, and the handover is fully automated. They've also been able to fully automate their web application firewall (WAF) in a pipeline.

| | |
|---|---|
| **500**<br>releases/month | **50%**<br>time savings |
| **50%**<br>reduction in code defects and reworks | **4x**<br>developer productivity |

Learn more

# Building joint solutions with a complete developer stack

## Panasonic Connect builds bridges across company divisions

"Since software assets, tools, and cloud know-how are individually optimized for each business division and system, it is difficult to understand what assets were available. We often fall into the trap of the 'reinvention of the wheel' in which similar functions are newly developed over and over again."

**Hisatoshi Adachi**

Executive IT Architect, R and D Division,
Panasonic Connect

Panasonic Connect, a subsidiary of the Japanese industry giant Panasonic Group, covers a wide range of businesses areas—including supply chain management, public services, infrastructure for daily life, and entertainment. After acquiring leading supply chain software company Blue Yonder, the company perceived that the differences in the two companies' development structures and cultures created a "sense of danger" across the different orgs. They set out to build a more collaborative development culture—facilitated by Microsoft Azure and GitHub.

# The challenges

### Collaboration concerns and siloed workflows

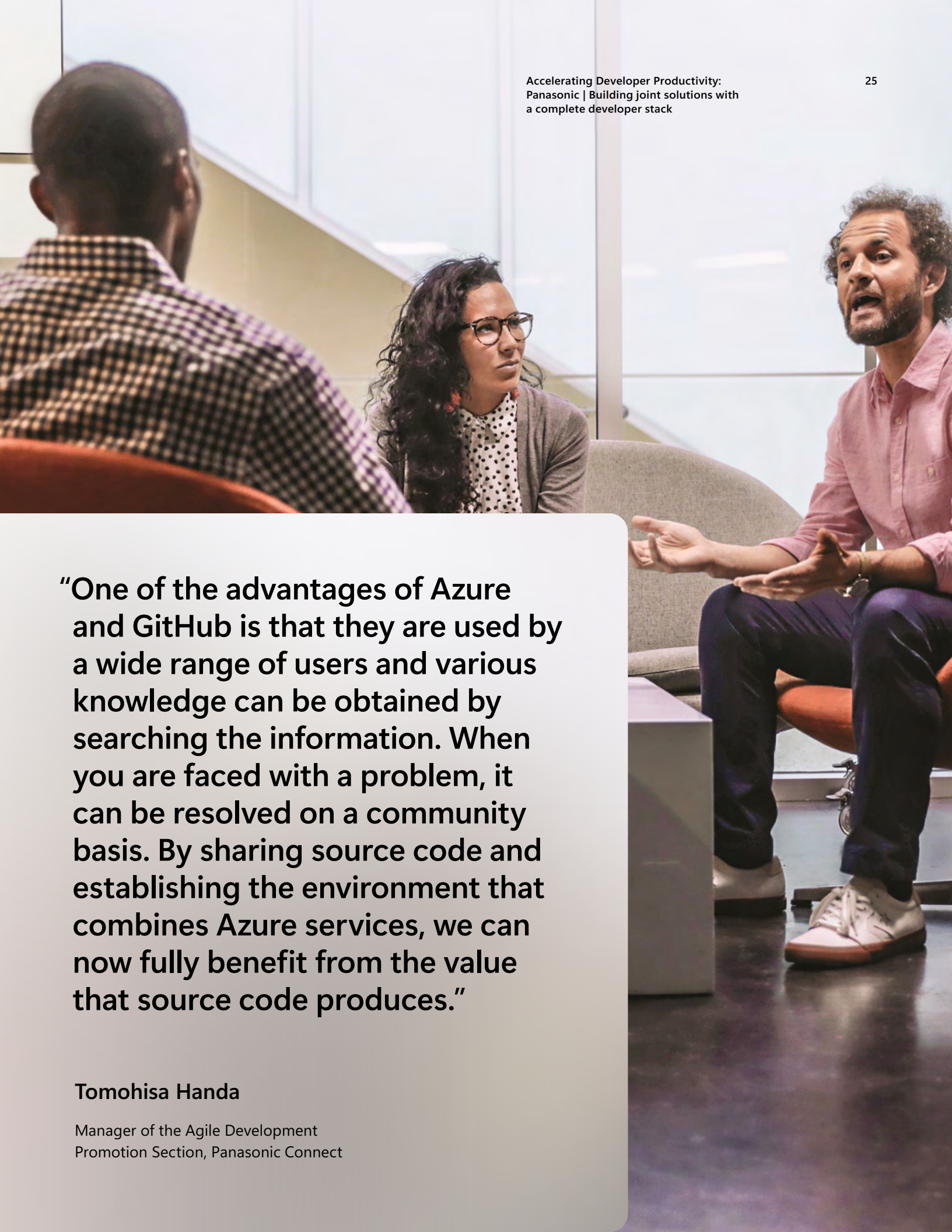
- There was a deep cultural divide between the developers from the storied, 100-year-old manufacturing giant and the 6-year-old cloud-based software company.
- The two branches struggled to bridge skill gaps related to cloud and open source.
- Siloed teams, tools, and resistance to sharing code created a crisis that leaders described as "a sense of danger."

### Productivity and time loss

- Simple deployments could take two or three days and needed to be tested manually.
- Waterfall-style development practices were clunky, highly manual, and hindered their ability to respond to sudden customer needs and market fluctuations.

### Tool sprawl, delayed deployments, and increased costs

- The two orgs struggled to standardize and streamline their development tools and IoT and edge technologies.
- Existing processes, primarily optimized for hardware production, lagged in the fast-paced cloud software release cycles.



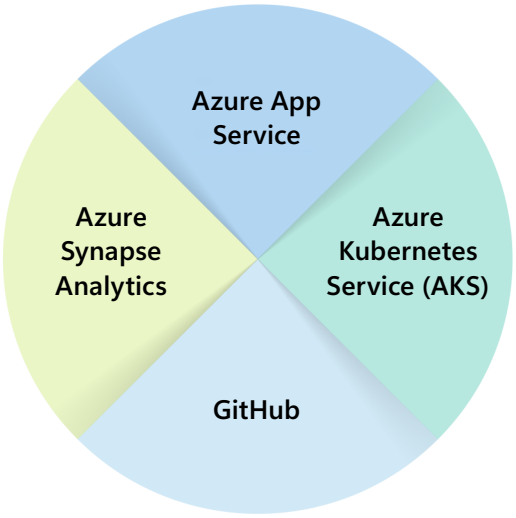
"One of the advantages of Azure and GitHub is that they are used by a wide range of users and various knowledge can be obtained by searching the information. When you are faced with a problem, it can be resolved on a community basis. By sharing source code and establishing the environment that combines Azure services, we can now fully benefit from the value that source code produces."

**Tomohisa Handa**

Manager of the Agile Development
Promotion Section, Panasonic Connect

# Solutions used

Leaders at Panasonic consider the successful collaboration initiatives, joint solutions, and boost in developer morale to be some of benefits of their adoption of Azure and GitHub solutions.




"Probably the biggest achievement was receiving feedback such as: 'I did not know it was so easy!' and 'It used to take two to three days to commit and deploy the code, including manually testing it. Now, it can be deployed in several minutes. I never want to go back to the way it was.'"

**Hisatoshi Adachi**

Executive IT Architect, R and D Division,
Panasonic Connect

# Outcomes

### 1. Created joint solutions using a full range of platform as a service (PaaS) offerings

Panasonic Connect was able to align their hardware development cycle with their software development cycle to deliver faster software updates using Microsoft Azure and GitHub. They created joint solutions by developing microservices for each module and function, and then linking them together.

The company used Azure PaaS services, including **Azure IoT Hub**, **Azure Synapse Analytics**, and **Azure Kubernetes Services (AKS)** to decrease the time to procure and build their system environment. This freed them from the burden of operating and maintaining a platform.

### 2. Accelerated development speed and improved efficiency with workflow automation

By improving efficiency and speed using **Azure Pipelines**, **Azure App Service**, and **GitHub** to build CI/CD pipelines, automate workflows, and make source code publicly available, Panasonic Connect was able to eliminate the need to "reinvent the wheel" across its development teams.

### 3. Fostered a culture of developer satisfaction and knowledge sharing

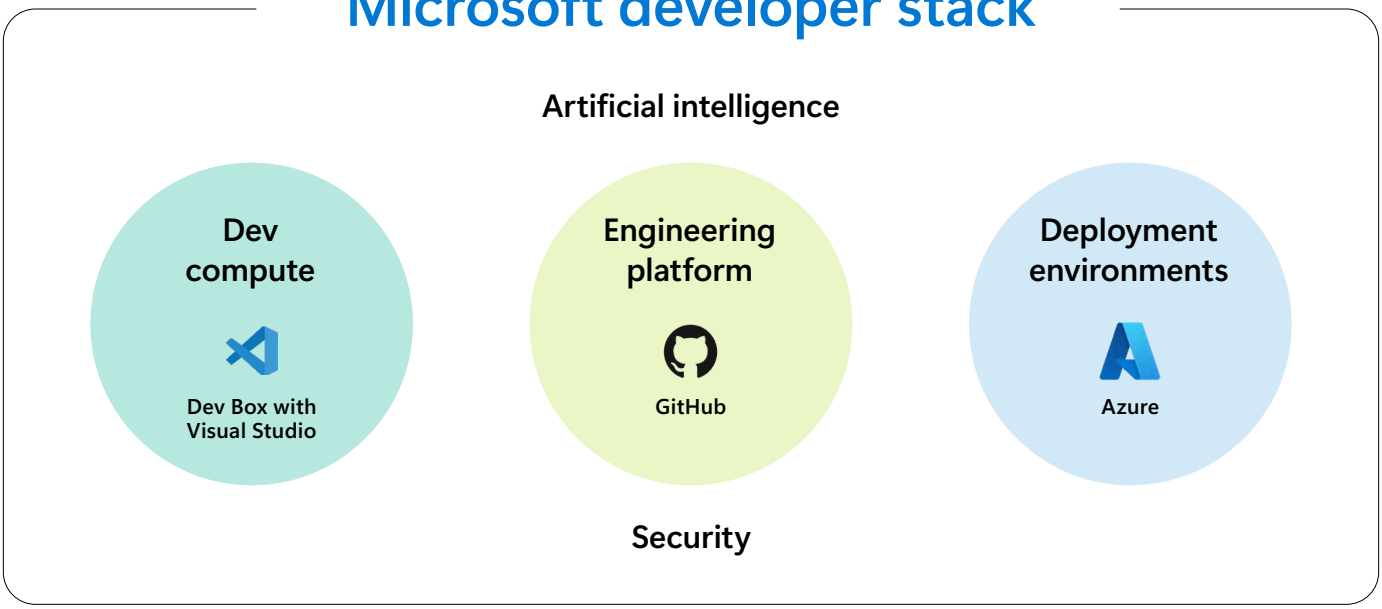
Developers across the company reported greater satisfaction and less frustration, thanks in part to the **GitHub community**. Using GitHub, they were able to conserve mental energy by solving problems more quickly with help from their peers.

Learn more

# Next steps to accelerate developer productivity

Bringing together the largest open-source community and some of developers' favorite coding and collaboration tools—GitHub, the Visual Studio family, and Azure—the **Microsoft developer stack** offers a comprehensive set of development tools to help your dev teams spend less time on code generation and remediation and more time on what they do best—creating.

## Microsoft developer stack




# Learn more

- Learn more about complete DevOps solutions with Microsoft Azure.
- Accelerate developer productivity with a GitHub Enterprise free trial.
- Learn how to deliver software faster and more securely by combining the practices and tools of GitHub and Azure.