FORRESTER®

GitHub Enterprise Cloud And Advanced Security Help Financial Services Organizations Produce Secure Code That Meets Auditing Requirements

Transactions in financial services organizations are often targeted by malicious actors. This increases the pressure to have a standardized approach to producing defect-free, secure software code. Since many financial services firms also offer technology services to their clients, it is essential that the software they provide meets both industry and regulatory standards.

Due to the pressure to get applications to market, coupled with the frequent shortage of developers, financial services organizations are forced to hire less experienced developers or developers who may not be familiar with the organization's coding practices. After the code is developed, security teams use various in-house developed methods to scan the code for security vulnerabilities or errors. Most organizations build code in silos, resulting in different security postures from department to department.

Local development, security, and operations (DevSecOps) teams often struggle to keep up with the constantly changing software patches and updates across thousands of code repositories. This decentralized approach exposes financial services organizations to security risks that often make it far into the software development lifecycle (SDLC). Remediating security defects that make it into production is costly and can threaten a financial services organization's reputation with its customers.

These problems are amplified as more financial services organizations move application development to the cloud. The very thing that makes open source attractive as a development platform also makes it easy for malicious actors to exploit.

Reported improvement in software developer productivity

Up to 22%

Total benefits present value (PV)

$136.80 million

Many financial services organizations implement the GitHub Enterprise Cloud development platform with GitHub Advanced Security to be able to build and ship secure software. The combination of GitHub Enterprise Cloud and Advanced Security ensures that software development teams orchestrate safe coding practices with agile software strategies. By building security into the software development workflow, this "end-to-end" security approach allows financial services organizations to identify security vulnerabilities in real-time.

GitHub commissioned Forrester Consulting to conduct a Total Economic Impact™ (TEI) study and examine the potential return on investment (ROI) enterprises may realize by deploying GitHub Enterprise Cloud and Advanced Security. To better understand the benefits, costs, and risks associated with this investment, Forrester interviewed four representatives and surveyed 147 respondents with experience using the GitHub set of solutions.

READ THE FULL STUDY

Forrester aggregated the interviewees' experiences and combined the results into a single composite organization, that is a firm with global operations and a central data center; this firm deploys the GitHub toolset in the cloud for more than 7,000 developers. Additionally, it uses GitHub Advanced Security, Actions, Codespaces, Discussions, Pages, and Copilot.

The full study aims to provide readers with a framework to evaluate the potential financial impact of GitHub Enterprise Cloud and Advanced Security on their organizations. However, this abstract will focus on the GitHub solution and the value it can provide to organizations.

INVESTMENT DRIVERS

Interviewees' and survey respondents' organizations chose to invest in GitHub Enterprise Cloud and Advanced Security for several reasons, including:

- **Need to increase productivity and maximize training effectiveness.** Before deploying GitHub Enterprise Cloud and Advanced Security, financial services organizations utilized various loosely coupled open-source tools to manage the CI/CD pipeline. Developers passed the code to testers, who manually scanned the code for security defects. If testers found security flaws, they sent the code back to the developers to rework and resubmit. DevSecOps had to manually track down all the code affected across many repositories if there were dependencies to other code. It could take days or weeks to remediate a single security defect. Another challenge was to train new developers to be productive as quickly as possible.

  An engineering director at a financial consulting firm said, "We didn't have good ways of measuring that, but the first month of an engineer joining was probably written off as non-effective for that engineer."

> **"We were not in a good place from a security perspective before GitHub. We did not have uniform ways for vulnerability dependency management. We did not have uniform ways of managing software vulnerability. We would do vulnerability management at the infrastructure layer."**
>
> *Engineering director, financial consulting*

- **Difficulty securing code.** Open-source software development made financial services organizations vulnerable to security breaches and incidents at every stage of the SDLC and production. Security was often an afterthought rather than embedded from the beginning of the SDLC. Since each team had its code repository, it was difficult for DevSecOps to ensure that security controls were implemented to protect the integrity of the code base across thousands of code repositories. As a result, security flaws often made it deep into the SDLC before being detected. The farther in the SDLC security flaws were discovered, the more difficult and costly they were to remediate.

> **"We are bound by industry frameworks like ISO 2001. We also have to meet some specific government requirements. Also, because we run client systems, we are bound by some client contractual requirements like NIST, CIS, and all of those."**
>
> *Engineering director, financial consulting*

- **Difficulty automating tasks.** Individual developers, DevOps, and DevSecOps teams manually managed complex security issues, usually with homegrown code. Each developer would tackle these problematic issues in different ways. Without automation, organizations relied on developers to remember to check for security flaws. Standardizing security code and automating its application across various tool stacks and repositories was complicated, resulting in an inconsistent security posture throughout the interviewees' organization.

> **"Previously, we were basically manually collecting the logs, and we sometimes lost an event or two."**
>
> *Head of defensive security, financial services*

- **Difficulty preparing for audits.** Financial services firms operate in a regulated environment. Developers not only have to meet internal coding standards but external regulatory standards as well. Given the increasing number of standards, it's complicated to ensure that all code passes the necessary audits. At interviewees' organizations, auditing functions were often separate from developer functions, and teams of auditors had to review the code to ensure it met various regulatory requirements. Preparing for these audits was time-consuming, and financial services organizations often had to keep years' worth of records and logs for auditors. Additionally, manual audit preparation left financial services organizations open to human error.

SOLUTION REQUIREMENTS

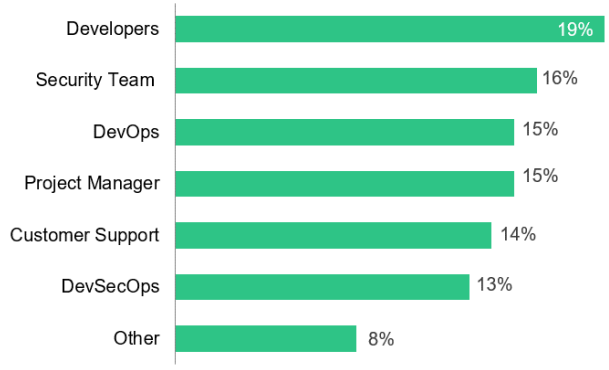
The interviewees' organizations were determined to reduce the frequency of security vulnerabilities deployed into production and speed up releasing software products to market. They searched for a solution that could:

- Reduce the number of security flaws.
- Better protect intellectual property.
- Increase developer productivity.
- Reduce the number of software defects.
- Reduce the amount of rework.
- Retire resources dedicated to legacy solutions.
- Speed up new-hire training.

Additionally, financial services organizations wanted to implement a single, intuitive platform for uninterrupted developer experience optimized to support collaboration across teams. They were looking to bring developers and non-developers together to collaborate on the same platform and allow for collaboration from anywhere users worked.

Survey participants provided their insights in the following graphic into the distribution of GitHub Enterprise Cloud users across their organizations.

"What percentage of your organization's GitHub users are the following?"
Note: Top 6 responses shown



Base: 143 GitHub Enterprise users
Source: A commissioned study conducted by Forrester Consulting on behalf of GitHub, August 2022

“We’ve prevented over 10,000 dependency vulnerabilities to be released to production environments.”

Head of defensive security, financial services

KEY QUANTIFIED RESULTS

GitHub Enterprise Cloud and Advanced Security enabled several benefits for the interviewees’ organizations. This section highlights four of the seven quantified benefits for their impact on the interviewees’ financial services organizations.

Increased developer productivity. When GitHub Actions was combined with GitHub Advanced Security, the financial services organizations discovered security flaws much sooner as GitHub’s code scanning capability scanned code and compared it against the GitHub Advisory Database. Alerts were then sent to developers if any security vulnerabilities were found. DevSecOps teams wrote workflows that automatically performed security scans and other tasks across multiple code repositories. Secret scanning scans repositories for known secret formats to prevent fraudulent use of credentials that were committed accidentally. These workflows were embedded into the GitHub Codespaces development environment, so when developers created new projects, the automated procedures were included from the beginning. As a result, developers spent less time writing code, moving code through the SDLC, and tracking security vulnerabilities.

One interviewee’s organization added Copilot to its GitHub toolbox. GitHub’s AI-driven software learned best practices and suggested them when developers were not following them. An engineering director at a financial consulting firm noted: “We are excited about Copilot, particularly the enterprise version of Copilot that can learn from code that you have within your own enterprise such that it makes your engineers more effective. For example, if you have someone writing a pipeline to deploy some code, Copilot can help that engineer copy a pipeline that already exists in our library, and they can use the time to focus on other things.”

Efficiencies from fewer code defects and vulnerabilities. Interviewees reported that without automation in their organization’s CI/CD environment, security vulnerabilities made it far into the SDLC before being detected. In many cases, the organizations did not discover the exposures until the code was deployed to production.

A head of defensive security at a financial services firm mentioned that financial services organizations often grow by acquisition and are under time pressure to absorb the acquired company’s code quickly while avoiding vulnerabilities. Before GitHub, discovering potential vulnerabilities in the code of a newly acquired organization was nearly impossible. They said: “Developers had some atrocious coding practices. That number will scare you; at one point, we found 100,000 security vulnerabilities.”

After implementing GitHub Enterprise Cloud and Advanced Security and utilizing Actions, Codespaces, Dependabot, and GitHub’s code and secret scanning capability, organizations could drastically reduce the number of issues. GitHub Dependabot alerted developers of security issues at every stage of the SDLC. Additionally, the tools quickly identified affected code dependencies in the software supply chain. This improved code quality and reduced the time developers spent troubleshooting defects.

The financial services organizations reduced the number of bugs developers created, eliminating tens of thousands of hours per year in remediation time.

Reduced developer onboarding time through automation. Interviewees said that before using GitHub Enterprise Cloud, their organizations' new developers needed to be trained on the proper coding methods and how to use the various tools they would need during the SDLC.

- After implementing GitHub, these financial services organizations drastically reduced training time for new developers. Since new developers typically generate more security vulnerabilities, GitHub Advanced Security and Dependabot proved helpful in catching security vulnerabilities early. This automation further reduced the time needed to train developers on best coding practices.
- GitHub Actions and Codespaces indirectly taught junior developers best practices by identifying poor coding practices.
- GitHub Discussions, a forum for sharing ideas, allowed new developers to ask questions of more senior developers, share best practices, and become part of the developer community.

An engineering director at a financial consulting firm confirmed: "We have many processes, and over 90% of developers don't know what's happening in the background. So, we do have a lot of automation, but if people had to do that manually — if they could figure out how to — that would take a long time."

> **"By automating everything, we could save our business approximately $2 million worth of work per year."**
>
> *Engineering director, financial consulting*

Return on investment:

433%

Time savings on IT audit preparation. The interviewees in the financial services sector explained that their organizations operate in a regulated environment. Not only do their developers have to meet internal coding standards, but they also have to follow external regulatory standards. Most financial regulations require a neutral group to perform the audits. Before GitHub, developers had to help auditors by producing the necessary documentation. Given the frequency of financial audits, this reduced the productivity of developers. A head of defensive security at a financial services organization said: "Some of our audits ask us for organizational charts, for example, roles and responsibilities and things like that. This documentation lives on those GitHub pages, so I can use this documentation to answer our audits."

These interviewees noted that GitHub Enterprise Cloud was able to automate the documentation process, cutting down the time developers need to spend manually documenting code. The standard documentation structures made it easier for auditors to find and compile the documentation necessary to be compliant. This helped the financial services organizations save time preparing for audits.

The full financial analysis aggregated the results in seven distinct quantifiable benefits and found that a composite organization experiences benefits of $168.42 million over three years versus costs of $31.63 million, adding up to a net present value (NPV) of $136.80 million and an ROI of 433%.

TOTAL ECONOMIC IMPACT ANALYSIS

For more information, download the full study: "The Total Economic Impact™ Of GitHub Enterprise Cloud and Advanced Security," a commissioned study conducted by Forrester Consulting on behalf of GitHub, November 2022.

STUDY FINDINGS

Forrester interviewed four representatives and surveyed 147 respondents at organizations with experience using GitHub Enterprise Cloud and Advanced Security and combined the results into a three-year composite organization financial analysis. Risk-adjusted present value (PV) quantified benefits include:

- **Increased developer productivity, saving $146.3 million.**
- **Reduced need to support local legacy and open source tools, valued $7.1 million.**
- **Efficiencies from fewer code defects and vulnerabilities, saving $5.2 million.**
- **Reduced developer onboarding training time through automation, valued at $3.2 million.**
- **Improved DevOps and site reliability engineer efficiency, enabling cost savings of $3.1 million.**
- **Time savings on IT audit preparation, valued at $2.7 million.**
- **Savings from retiring legacy development tools, enabling cost savings of $811,900.**

Return on investment (ROI)

433%

Net present value (NPV)

$136.8M

READ THE FULL STUDY


FORRESTER®