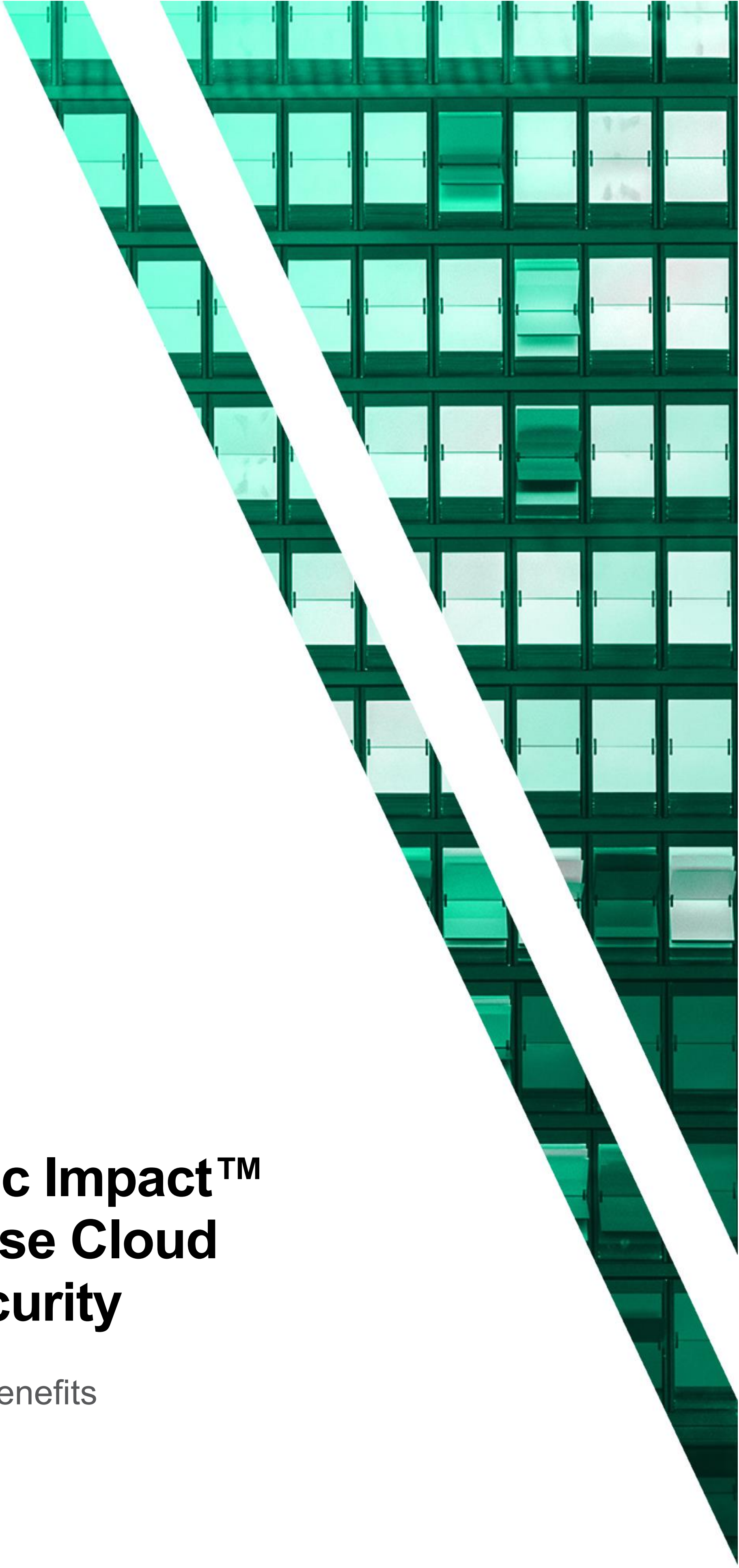
FORRESTER®



The Total Economic Impact™ Of GitHub Enterprise Cloud And Advanced Security

Cost Savings And Business Benefits Enabled By GitHub

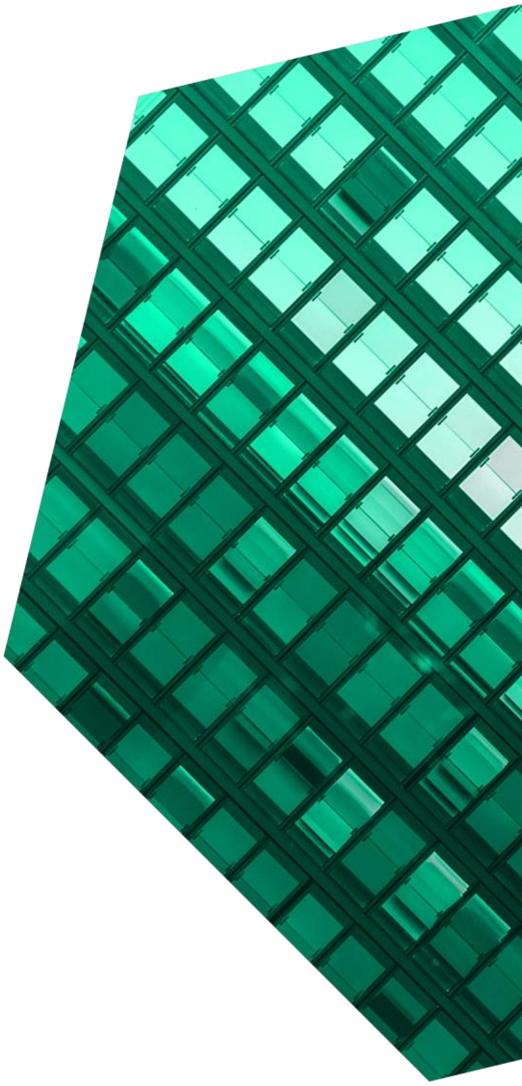
NOVEMBER 2022

A FORRESTER TOTAL ECONOMIC IMPACT™ STUDY COMMISSIONED BY GITHUB

Table Of Contents

Consulting Team: Jonathan McKinney

Anna Orbán-Imreh



ABOUT FORRESTER CONSULTING

Forrester provides independent and objective research-based consulting to help leaders deliver key transformation outcomes. Fueled by our customer-obsessed research, Forrester’s seasoned consultants partner with leaders to execute on their priorities using a unique engagement model that tailors to diverse needs and ensures lasting impact. For more information, visit forrester.com/consulting.

Executive Summary

Organizations are under increasing pressure to produce secure, defect-free code. The environment is often a patchwork of open source tools from different vendors connected by homegrown code. Moving code between different tools can lead to code defects. Remediating defects is very costly and can threaten an organization's reputation. GitHub Enterprise Cloud and Advanced Security solve this by providing a single uniform platform for developing secure code at all phases of the software development life cycle.

GitHub provides an enterprise-grade development platform that helps organizations of all sizes plan, build and ship secure software. The combination of GitHub Enterprise Cloud, Codespaces, Advanced Security and Actions ensures software organizations orchestrate DevSecOps and Agile software strategies, which accelerates time to market for innovative customer experiences.

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. The purpose of this study is to provide readers with a framework to evaluate the potential financial impact of GitHub Enterprise Cloud and Advanced Security on their organizations.

To better understand the benefits, costs, and risks associated with this investment, Forrester interviewed four representatives and surveyed 147 respondents with experience using GitHub Enterprise Cloud and Advanced Security. For the purposes of this study, Forrester aggregated the experiences of the interviewees and survey respondents into a single composite organization that is a global consulting firm with 120,000 employees and a revenue of $42 billion per year.

These interviewees noted that prior to using GitHub Enterprise Cloud and Advanced Security, their organizations managed their code quality and developer productivity by utilizing a combination of commercial and homegrown open source tools. Often there were different tool stacks from department to department with little collaboration or code sharing. It took teams of testers to scan code manually for defects and security vulnerabilities. This decentralized approach left the organizations with poor code quality, low developer productivity, silos of legacy tools, and exposure to security risks.

KEY STATISTICS

Return on investment (ROI)
433%

Net present value (NPV)
$136.80M

After investing in GitHub Enterprise Cloud with Advanced Security, the interviewees' organizations saw many improvements, including better performance, improved code quality, improved compliance, and enhanced security.

> **"A huge advantage of GitHub Advanced Security is their ability to detect passwords and credentials in code."**
>
> *Senior principal engineer, technology*

KEY FINDINGS

Quantified benefits. Three-year, risk-adjusted present value (PV) quantified benefits for the composite organization include:

- **Increased developer productivity.** GitHub Enterprise Cloud, Advanced Security and Actions manage the movement of code through the SDLC by automating tasks usually performed manually by developers, DevOps, and DevSecOps. GitHub Codespaces is a cloud-based development environment that creates templates, enforces standardization, and reduces developers' time by packaging standard, repeatable code. GitHub Discussions is a collaborative forum where developers can share best practices that improve their efficiency. As a result of leveraging these capabilities and processes, the composite organization's developers realize productivity gains of 12% in Year 1 to 22% in Year 3, resulting in savings of $146.3 million.

- **Reduced need to use and support local legacy and open source tools.** By centralizing development and sharing code using cloud-based GitHub Codespaces, the composite organization reclaims hours of time developers previously spent managing and updating open source software, in-house code-scanning tools, and the corresponding infrastructures. The composite organization realizes productivity gains of 25% in Year 1 to 75% in Year 3, resulting in savings of $7.1 million.

  Efficiencies from fewer code defects and vulnerabilities. GitHub Actions and GitHub Advanced Security reduces the number of security vulnerabilities and risk of leaked secrets by automatically scanning code. GitHub Dependabot alerts developers of known vulnerabilities in their open source dependencies at every stage of the SDLC. Developers are notified immediately in their workflow when there is a security issue, and the tools also quickly identify how to remediate the issue. This improves code quality and reduces the time developers spend troubleshooting defects. The composite organization realizes an overall reduction in the time spent on remediation, resulting in savings of $5.2 million.

- **Reduced developer onboarding training time through automation.** GitHub Actions automates many tasks a developer would typically need to be trained to do. GitHub Codespaces provides prebuilt environments, reducing the time a developer needs to write standard, repeatable code. By providing developers with a head start on projects, the composite organization no longer needs to train new hires to perform these tasks. New-hire training in the composite organization is reduced from 10 days to two days, resulting in savings of $3.2 million.

- **Improved DevOps, DevSecOps and site reliability engineer efficiency.** GitHub Actions, Codespaces, and Advanced Security reduce the time DevOps and DevSecOps teams spend managing the centralized developer environment through the creation of templates and workflows. The standardization and automation also reduce the number of software defects and security flaws DevOps and DevSecOps teams need to remediate. The composite organization sees a reduction in time spent managing the environment and remediating issues from 8% in Year 1 to 15% in Year 3, resulting in savings of $3.1 million.

- **Time savings on IT audit preparation.** GitHub Pages automates the documentation process, cutting down the time developers need to spend manually documenting code. The self-serve standard documentation structures make it easier for auditors to find and compile the documentation necessary to be compliant without having to interrupt developers. This helps the composite organization save time preparing for audits, and it is able to cut the time auditors spend preparing for audits in half, resulting in savings of $2.7 million.

- **Savings from retiring legacy development tools.** GitHub Enterprise Cloud, Actions, and Advanced Security perform most of the centralized tasks that legacy open source tools perform, eliminating the need for these tools and reducing the friction caused by moving code between the remaining tools with standard application programming interfaces (APIs). Standardizing GitHub Enterprise Cloud and Advanced Security allows the composite organization to eliminate centralized legacy tools and the hardware that supports them. The composite organization realizes a total cost of ownership savings of $811,900.

> **"We wanted to empower developers to use the tools they prefer while also creating a common language across all teams. By standardizing on GitHub, our developers can work the way they want to while still making it easy to share and discover each other's code."**
>
> *Senior principal engineer, technology*

Unquantified benefits. Benefits that provide value for the composite organization but are not quantified for this study include:

- **Employee satisfaction.** The use of GitHub saves the composite organization's staff time and energy, and it allows them to spend more time on productive activities. This increases employee satisfaction and retention of highly skilled resources in several IT departments.
- **Customer satisfaction.** With fewer defects in its software code, the composite organization sees less application downtime, while faster mean time to remediation (MTTR) for problems that still make it into production increases application uptime and overall customer satisfaction.
- **Better collaboration between teams.** GitHub Enterprise Cloud enables the development team to centralize the enterprise code base, making it easier for developers in different departments to share best practices with the organization. For the security teams, GitHub Advanced Security allows the organization to dynamically manage and react to security threats with a unified security posture. The composite organization experiences enhanced collaboration among developers across regions and teams.

Costs. Three-year, risk-adjusted PV costs for the composite organization include:

- **Fees to GitHub.** Each user of GitHub requires a separate license. The composite organization incurs annual license fees and professional services fees payable to GitHub totaling $16.8 million.
- **Internal Costs**. The composite organization incurs a cost for experts in its DevOps and DevSecOps departments configuring and customizing GitHub and training developers on how to use it. The initial and ongoing customization and training costs total $14.8 million.

The financial analysis which is based on the interviews and survey 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%.

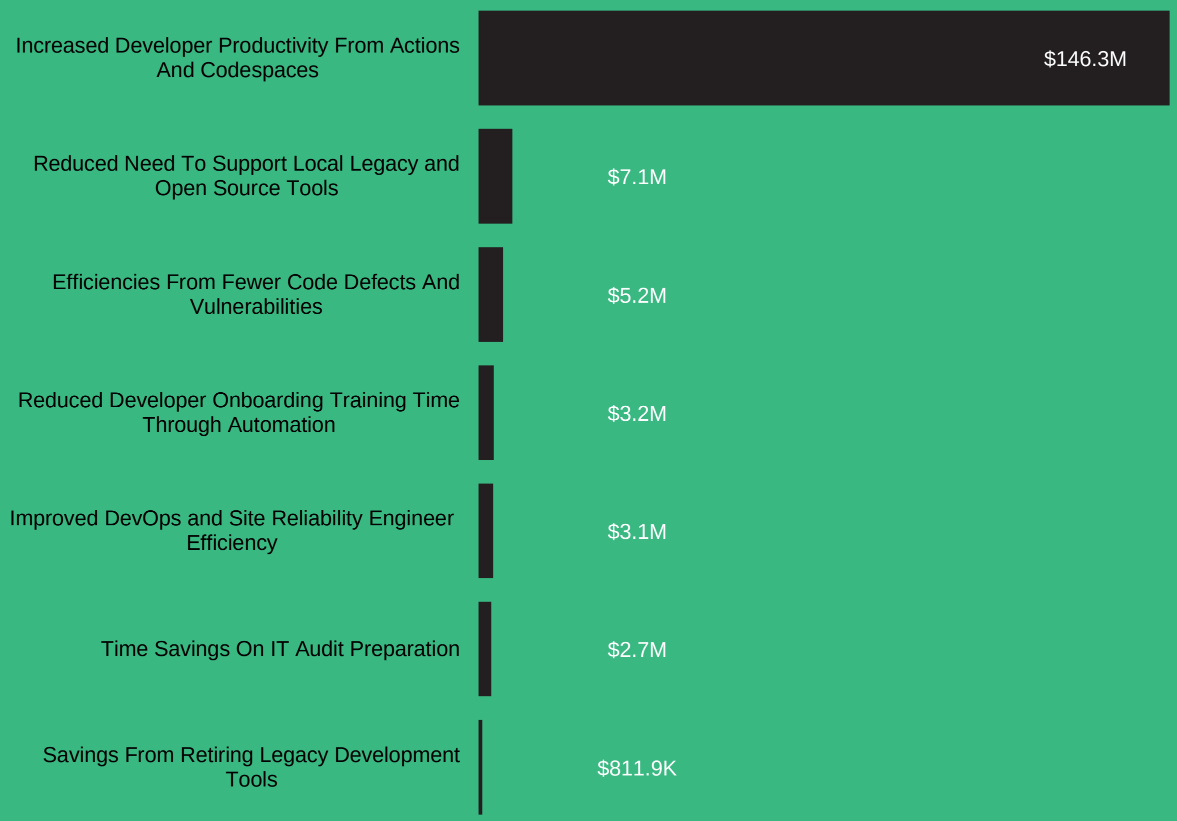
ROI
433%

BENEFITS PV
$168.42M

NPV
$136.80M

PAYBACK
<6 months

Benefits (Three-Year)



> “With GitHub, we are able to focus our engineers on building much more self-service management capability rather than having dedicated teams that execute certain management tasks on behalf of others who are consuming the service.”
>
> *Director of architecture, telecom*

TEI FRAMEWORK AND METHODOLOGY

From the information provided in the interviews and survey, Forrester constructed a Total Economic Impact™ framework for those organizations considering an investment in GitHub Enterprise Cloud and Advanced Security.

The objective of the framework is to identify the cost, benefit, flexibility, and risk factors that affect the investment decision. Forrester took a multistep approach to evaluate the impact that GitHub can have on an organization.

DUE DILIGENCE

Interviewed GitHub stakeholders and Forrester analysts to gather data relative to GitHub Enterprise Cloud and Advanced Security.

INTERVIEWS AND SURVEY

Interviewed four representatives and surveyed 147 respondents at organizations using GitHub Enterprise Cloud and Advanced Security to obtain data with respect to costs, benefits, and risks.

COMPOSITE ORGANIZATION

Designed a composite organization based on characteristics of the interviewees and survey respondents.

FINANCIAL MODEL FRAMEWORK

Constructed a financial model representative of the interviews and survey using the TEI methodology and risk-adjusted the financial model based on issues and concerns of the interviewees and survey respondents.

CASE STUDY

Employed four fundamental elements of TEI in modeling the investment impact: benefits, costs, flexibility, and risks. Given the increasing sophistication of ROI analyses related to IT investments, Forrester's TEI methodology provides a complete picture of the total economic impact of purchase decisions. Please see Appendix A for additional information on the TEI methodology.

DISCLOSURES

Readers should be aware of the following:

This study is commissioned by GitHub and delivered by Forrester Consulting. It is not meant to be used as a competitive analysis.

Forrester makes no assumptions as to the potential ROI that other organizations will receive. Forrester strongly advises that readers use their own estimates within the framework provided in the study to determine the appropriateness of an investment in GitHub.

GitHub reviewed and provided feedback to Forrester, but Forrester maintains editorial control over the study and its findings and does not accept changes to the study that contradict Forrester’s findings or obscure the meaning of the study.

GitHub provided the customer names for the interviews but did not participate in the interviews.

Forrester fielded the double-blind survey using a third-party survey partner.

The GitHub Enterprise Cloud & Advanced Security Customer Journey

Drivers leading to the investment in GitHub Enterprise Cloud and Advanced Security

KEY CHALLENGES

Before GitHub, the interviewees' organizations used many different open source solutions across the organizations. Each legacy solution required custom coding to interface with other legacy solutions within the organization. Managing code between the developers and testers to ensure that code was defect-free throughout the SDLC involved multiple teams and manual steps, and it added weeks to the SDLC process. This method of deploying code to production also introduced human error.

As a common theme, interviewees noted that with a constantly evolving security landscape and new vulnerabilities, their organizations' developers and DevSecOps experts spent considerable time updating the code and scanning for these emerging security vulnerabilities. In a way, developers became part-time DevSecOps and were required to make decisions about security that should have been left to the experts. Complicated code with multiple dependencies was challenging to scan thoroughly, which resulted in delays in getting software products to market and made it easier for security vulnerabilities to get into production.

Both interviewees and survey respondents noted how their organizations struggled with common challenges, including:

- **Frequent quality issues.** All respondents said the number one challenge for their organization was producing quality code. Large organizations tend to create silos of development, each with its distinct way of building code. Departments would often buy their software and servers and manage them locally. This led to using many open source tools, each designed for a specific purpose but not easily built to interface with other open source tools in the organization. This lack of visibility and collaboration between silos made it difficult for DevOps and DevSecOps to manage the SDLC for the entire organization. Often, they had to create intermediate code to manage context switching between different software tools. For all the reasons above, maintaining quality across the tool stacks with multiple open source tools, homegrown scripts, and lack of collaboration was challenging.

- **Difficulty securing code.** Security was the next biggest challenge. Open source software development made the organizations potentially vulnerable to security breaches and incidents at every stage of the SDLC and production. Security was often an afterthought rather than being embedded throughout the SDLC. Since each team had its own code repository, that made it difficult for the security team to ensure that security controls were implemented to protect the integrity of the code base. A deputy lead of dynamic platforms at an automotive manufacturer described: "Every larger project or area had their own way of storing source code. There was no regulation to that. Everybody could use more or less what they liked."

  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.

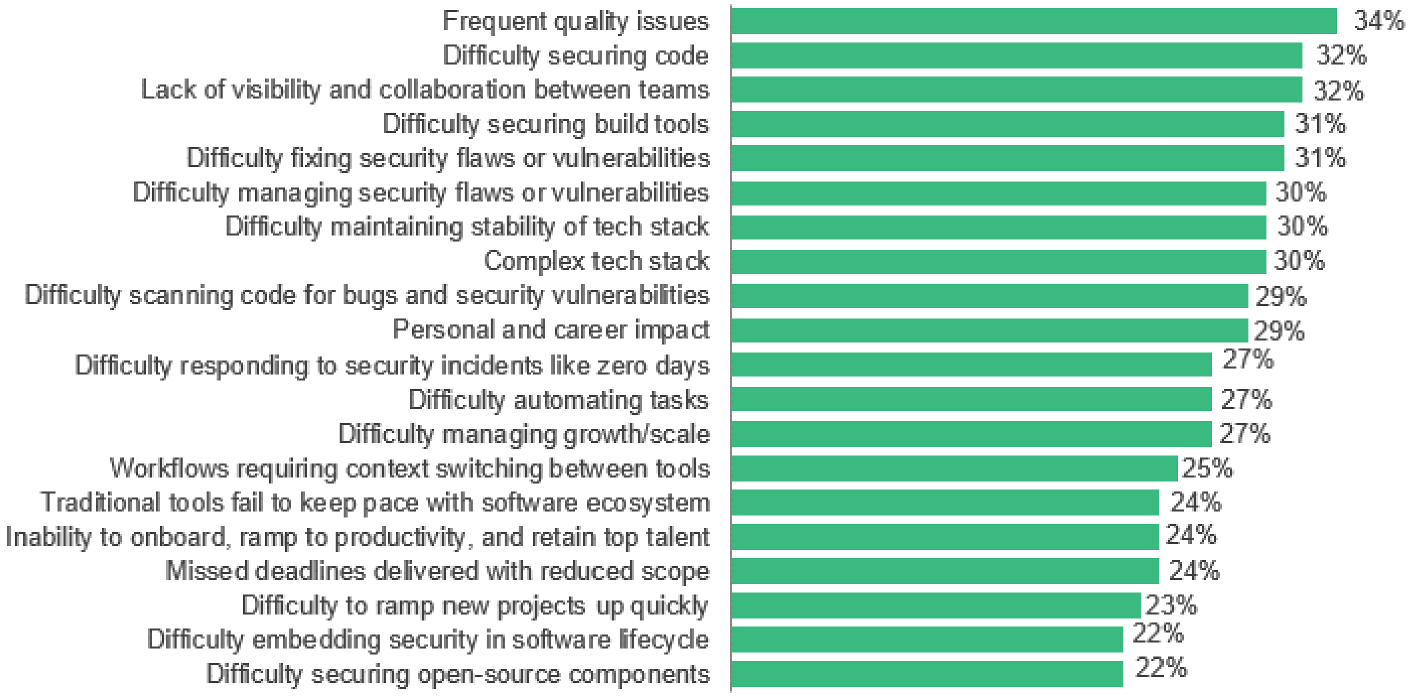
> **"Even if development teams had DevOps methodologies, they were each locally optimized at the segment level or, worst case, they were optimized locally at the team level. So, every team was maybe writing their own. Local optimization is death by a thousand paper cuts."**
>
> *Senior principal engineer, technology*

- **Difficulty automating tasks.** Individual developers, DevOps, and DevSecOps teams manually managed complex quality and security issues, usually with homegrown code. Each developer would tackle these problematic issues in different ways. Standardizing code and automating its application across various tool stacks was complicated, resulting in further code quality and security issues for the interviewees' organizations.
- **Difficulty ramping up new projects quickly.** The silos of different development tools and processes, nonstandard code bases throughout the organizations, and pressures to decrease time to market led to complexities that slowed the development process. This was especially true for junior developers who often would "reinvent the wheel" for each new software project. The deputy lead of dynamic platforms at the automotive manufacturing firm said: "We were very distributed. We were defragmented to many departments [and] many tools."

> **"We consider source code an important asset and part of our intellectual property. So, one big driver to consider GitHub was our strategy to bring all source code together to have everything documented at one central location and to also have it under our control."**
>
> *Deputy lead of dynamic platforms, automotive manufacturing*

Areas of improvement. Survey respondents reported the following challenges their organizations were looking to resolve through the investment in GitHub:

"Which of the challenges did your organization hope to solve or improve by investing in GitHub?"



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

SOLUTION REQUIREMENTS

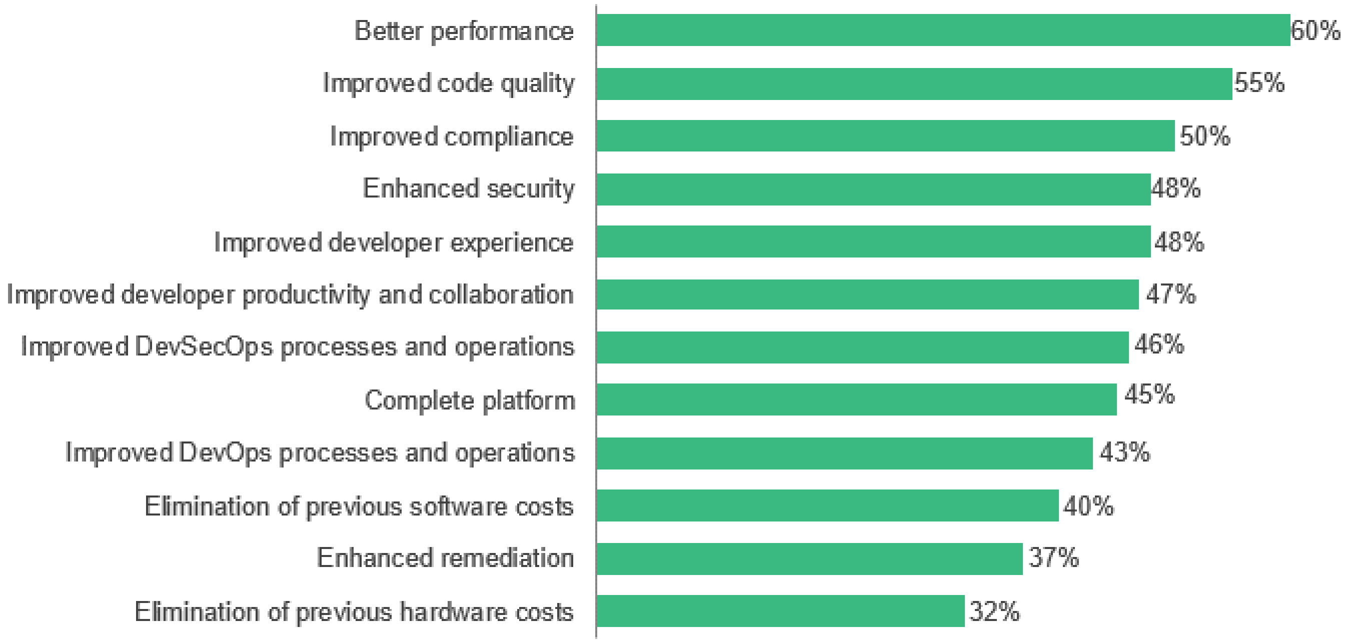
Investment objectives. The interviewees' organizations were determined to reduce the frequency of security vulnerabilities getting into production and speed up releasing software products to market. They searched for a solution that could:

- 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.

Criteria for platform selection. Survey respondents ranked the following factors that influenced the decision-making process for their organizations when evaluating and selecting GitHub:

> **"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*

"Before investing in GitHub, what were you looking for in a solution?"



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

COMPOSITE ORGANIZATION

Based on the interviews and survey, Forrester constructed a TEI framework, a composite company, and an ROI analysis that illustrates the areas financially affected. The composite organization is representative of the four interviewees and the 147 survey respondents, and it is used to present the aggregate financial analysis in the next section. The composite organization has the following characteristics:

Description of composite. The composite organization is a consulting enterprise with 120,000 employees and 7,000 developers, growing at 500 per year. It has global operations and a central data center.

Deployment characteristics. The composite organization deploys GitHub Enterprise Cloud in the cloud, it uses GitHub Advanced Security, Actions, Codespaces, Discussions, and Pages, and it is testing Copilot. The development team manages 20,000 code repositories and $100,000 in legacy hardware. It has used GitHub for five years.

Key Assumptions

- **Global operations with central data center**
- **GitHub Enterprise Cloud and Advanced Security deployed in the cloud**
- **7,000 GitHub users in Year 1, adding 500 users per year**
- **20,000 code repositories**
- **Uses GitHub Codespaces, Actions, Dependabot, and Discussions**
- **Testing GitHub Copilot**

> **“We had many silos and many individual teams that would create their own versions of software or other kind of building blocks that would repeat functionality without being able to discover reusable code bases or building blocks.”**
>
> — Engineering director, financial consulting

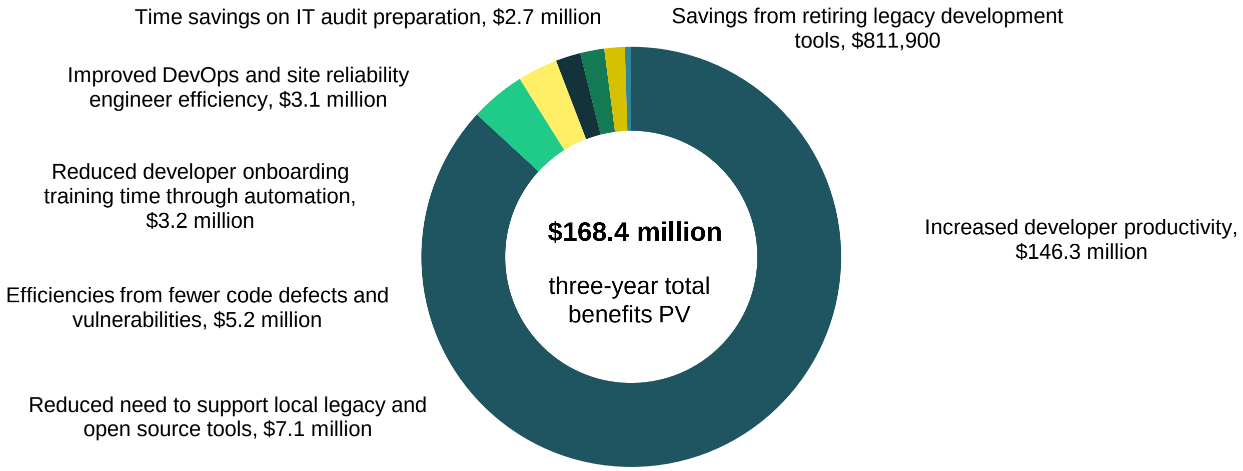
Analysis Of Benefits

Quantified benefit data as applied to the composite

Total Benefits

Ref.	Benefit	Year 1	Year 2	Year 3	Total	Present Value
Atr	Increased developer productivity	$36,893,961	$65,965,185	$77,484,479	$180,343,625	$146,271,882
Btr	Reduced need to support local legacy and open source tools	$1,356,395	$2,910,229	$4,661,500	$8,928,124	$7,140,488
Ctr	Efficiencies from fewer code defects and vulnerabilities	$1,957,950	$2,100,450	$2,242,950	$6,301,350	$5,201,025
Dtr	Reduced developer onboarding training time through automation	$714,000	$1,530,000	$1,734,000	$3,978,000	$3,216,334
Etr	Improved DevOps and site reliability engineer efficiency	$876,096	$1,314,144	$1,642,680	$3,832,920	$3,116,690
Ftr	Time savings on IT audit preparation	$1,071,000	$1,071,000	$1,071,000	$3,213,000	$2,663,418
Gtr	Savings from retiring legacy development tools	$237,500	$332,500	$427,500	$997,500	$811,890
	Total benefits (risk-adjusted)	$43,106,902	$75,223,508	$89,264,108	$207,594,519	$168,421,727

BENEFITS BY CATEGORY



This section examines seven quantified benefits and provides insight into the data points and evidence collected during the customer interviews and survey as well as the underlying models and assumptions used in the financial analysis for this use case.

INCREASED DEVELOPER PRODUCTIVITY

Evidence and data. Before deploying GitHub Enterprise Cloud, interviewees' organizations utilized various loosely coupled open source tools to manage their SDLCs. Because of this complex stack, developers had to write homegrown solutions to address the context switching between one open source tool to another. Developers passed code to testers, who would manually scan the code for defects. If testers found flaws, they would send the code back to the developers to rework and resubmit the code. If there were dependencies to other code, developers had to manually track down all the code that was affected. It could take days or weeks to remediate a single defect.

Security was an afterthought, which resulted in discovering security flaws late in the SDLC or after the code was deployed into production. Given the insufficient supply of developers, many junior-level developers were responsible for writing their own scanning tests, resulting in poorly written code and exposure to security vulnerabilities. In general, there was a lack of modern continuous integration/continuous delivery (CI/CD) with integrated security.

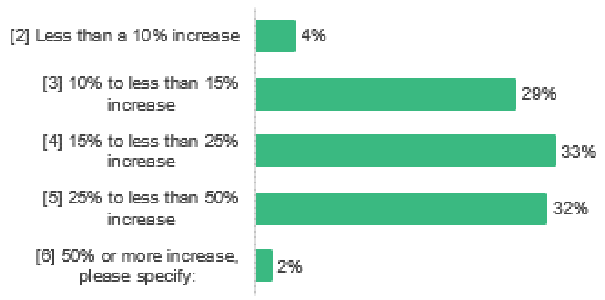
Once development teams installed GitHub Enterprise Cloud and Advanced Security, they could automate many of these manual processes with event-driven workflows using GitHub Actions. When combined with GitHub Advanced Security, the organizations discovered security flaws much sooner because GitHub's secret scanning capability scanned code and compared it against the GitHub Advisory Database. Alerts were then sent to developers if any security vulnerabilities were found.

DevOps and DevSecOps teams wrote Actions workflows that automatically performed scans and other tasks. These workflows were embedded into Codespaces, so when developers created new projects, the automated procedures were included from the beginning. As a result, developers spent less time writing code, less time moving code through the SDLC, and less time tracking software defects and security vulnerabilities.

> **"Luckily, with GitHub Actions, we created a lot of automation that allowed people to do self-service."**
>
> *Senior principal engineer, technology*

Survey participants noted the following productivity improvements:

"By what percentage did GitHub increase developer productivity for your organization?"



Base: 115 – GitHub Enterprise Cloud users who said it increased developer productivity at their organization
Source: A commissioned study conducted by Forrester Consulting on behalf of GitHub, August 2022

Modeling and assumptions. Based on the GitHub customer interviews, Forrester estimates the following about the composite organization:

- Of its 7,000 GitHub license users, 6,870 are developers who write code, while 130 are DevOps and DevSecOps who write administrative code.
- The fully loaded hourly cost of a developer FTE is $75.

- A developer working 2,080 hours a year spends 75% of that time on software development tasks. Of these 1,560 hours, 85% (1,326 hours) are spent coding, scanning, and on rework.
- After implementing GitHub Enterprise Cloud and Advanced Security, the composite organization sees productivity gains of 12% in Year 1. The initial investment in creating templates and workflows by DevOps and DevSecOps contributes to accumulative productivity gains of 20% in Year 2 and 22% in Year 3.
- The composite organization sees a 50% productivity recapture rate.

Risks. The benefit can vary by organization due to the following factors:

- The number of developers in the organization using GitHub.
- The level of experience the developers have with GitHub and with writing code in general.
- The level of automation the DevOps and DevSecOps teams implement with GitHub Actions and Codespaces.
- The utilization of GitHub Advanced Security.
- The level of democratization of access to code throughout the organization.
- The number of legacy tools in production.

> **"Dependabot opens an issue, for example: for dependency bugs. Earlier, it was an infinite wait time because it wasn't being addressed. Now, it goes from infinite to essentially addressable as soon as the developer can get to it. So, the ROI there is huge."**
>
> *Director of architecture, telecom*

Results. To account for these risks, Forrester adjusted this benefit downward by 10%, yielding a three-year, risk-adjusted total PV (discounted at 10%) of over $146.3 million.

Increased Developer Productivity

Ref.	Metric	Source	Year 1	Year 2	Year 3
A1	Number of developers	Composite	6,870	7,370	7,870
A2	Hourly fully loaded cost of a developer FTE	TEI standard	$75	$75	$75
A3	Hours worked on coding, scanning and rework	Composite	1,326	1,326	1,326
A4	Productivity improvements with GitHub	Findings	12%	20%	22%
A5	Productivity realization factor	TEI standard	50%	50%	50%
At	Increased developer productivity	A1*A2*A3*A4*A5	$40,993,290	$73,294,650	$86,093,865
	Risk adjustment	↓10%			
Atr	Increased developer productivity (risk-adjusted)		$36,893,961	$65,965,185	$77,484,479
	Three-year total: $180,343,625		**Three-year present value: $146,271,882**		

REDUCED NEED TO SUPPORT LOCAL LEGACY AND OPEN SOURCE TOOLS

Evidence and data. Before implementing GitHub, each of the organizations' developer teams would determine the tools they used to create code. There was little coordination on tool selection between teams. Because of the decentralization, the DevOps and DevSecOps teams were not able to support these diverse systems. Each developer team would have to assign a developer as the local DevOps person responsible for maintaining the local toolset and infrastructure. This resource would perform all the functions that the centralized DevOps team would perform but without the institutional knowledge base of the data center. This developer also became a bottleneck should an incident occur within the local development environment.

Team productivity depended on the local developer and their ability to resolve system issues quickly. A senior principal engineer at a technology firm noted the problems when a local system went down. They said: "We waited 1 hour, 2 hours, 5 hours, or even 10 for the developer to wake up. That's lost time on the whole team that cannot build, that cannot deploy, etc. Without these issues, I would say that the productivity increase for the team can easily be 30% to 40%."

After installing GitHub, the time local DevOps developers spent managing this environment was eliminated. The developers returned to their teams to focus solely on productive coding. DevOps and DevSecOps handled problems centrally, reducing the downtime for developers.

Modeling and assumptions. Based on the GitHub customer interviews, Forrester estimates the following about the composite organization:

- By centralizing development and standardizing code using GitHub Codespaces, the composite organization eventually eliminates hundreds of local third-party servers and associated tools licenses, reclaiming hours of developer time spent managing these environments.
- Of the 6,870 software developers, 10% (687 developers) are assigned to managing distributed legacy applications. They spend 15% (234 hours) of their productive time managing the local server and legacy software.
- The fully loaded hourly cost of a developer FTE is $75.
- After implementing GitHub Enterprise Cloud and Advanced Security, the composite organization phases out the maintenance of legacy tools and saves 25% on maintenance labor in Year 1, 50% in Year 2, and 75% in Year 3.
- The composite organization sees a 50% productivity recapture rate.

> **“With GitHub, we no longer have to support infrastructure services and operating systems, manage patch vulnerability, and do all of the things that you need to do when running your own infrastructure.”**
>
> *Engineering director, financial consulting*

Risks. The benefit can vary by organization due to the following factors:

- The number of local developers performing DevOps tasks.
- The number of legacy tools that can be retired.
- The speed at which legacy tools and servers can be decommissioned.

Results. To account for these risks, Forrester adjusted this benefit downward by 10%, yielding a three-year, risk-adjusted total PV (discounted at 10%) of $7.1 million.

Reduced Need To Support Local Legacy And Open Source Tools

Ref.	Metric	Source	Year 1	Year 2	Year 3
B1	Number of local operations developers	Composite	687	737	787
B2	Hours spent on local operations per developer (annually)	Composite	234	234	234
B3	Productivity gains from fewer legacy tools requiring maintenance	Interviews	25%	50%	75%
B4	Hourly fully loaded cost of a developer FTE	TEI standard	$75	$75	$75
B5	Productivity realization factor	TEI standard	50%	50%	50%
Bt	Reduced need to support local legacy and open source tools	B1*B2*B3*B4*B5	$1,507,106	$3,233,588	$5,179,444
	Risk adjustment	↓10%			
Btr	Reduced need to support local legacy and open source tools (risk-adjusted)		$1,356,395	$2,910,229	$4,661,500
	Three-year total: $8,928,124		Three-year present value: $7,140,488		

EFFICIENCIES FROM FEWER CODE DEFECTS AND VULNERABILITIES

Evidence and data. Interviewees reported that without automation in their organization's CI/CD environment, software defects and security vulnerabilities made it far into the SDLC before being detected. In many cases, the exposures were not discovered until the infrastructure layer.

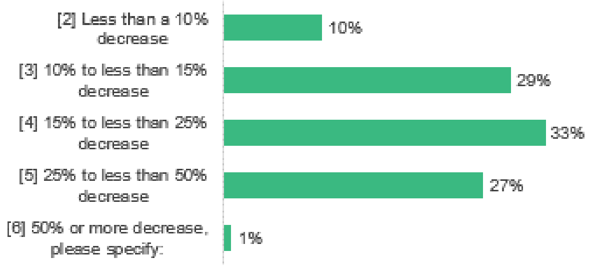
After implementing GitHub Enterprise Cloud and Advanced Security and utilizing Actions, Codespaces, Dependabot, and GitHub's secret scanning capability, the number of issues was drastically reduced. A director of architecture at a telecom firm said: "When we turned on Advanced Security, we immediately found about 15,000 known issues, and we've cut that down to about 5,000. So, it led to pretty rapid attention to things that were largely unknown previously."

GitHub Actions and Advanced Security reduced the number of software bugs and security vulnerabilities by automatically scanning code. GitHub Dependabot alerted developers of code defects and security issues at every stage of the SDLC. Additionally, the tools quickly identified what other code was affected. This improved code quality and reduced the time developers spent troubleshooting defects.

Interviewees' organizations reduced the number of bugs developers created, eliminating tens of thousands of hours per year in remediation time. The director of architecture at the telecom firm said: "Previously, it took two developers on average a week — or 80 hours — to address an issue. Now, it takes them 6 hours altogether. So, that's a big drop there [and] a pretty significant improvement."

Survey participants noted the following reduction of security flaws in their organizations due to GitHub:

"By what percentage did GitHub decrease the number of security flaws for your organization?"



Base: 109 – GitHub Enterprise Cloud users who said it reduced the number of security flaws at their organization
Source: A commissioned study conducted by Forrester Consulting on behalf of GitHub, August 2022

Modeling and assumptions. Based on the GitHub customer interviews, Forrester estimates the following about the composite organization:

- The composite organization's 6,870 developers finds an average of 38,667 code defects per year.
- After implementing the GitHub solution stack, each of the composite's developers reduces the number of defects that need remediation by one defect per year on average, for a total of 6,870 defects.
- The industry standard estimation for MTTR is 8 hours.
- The fully loaded hourly cost of a developer FTE is $75.
- The composite organization sees a 50% productivity recapture rate.

Risks. The benefit can vary by organization due to the following factors:

- The amount of code created by developers.
- The effective use of the GitHub solution stack in implementing automation in the CI/CD environment.
- The number of legacy open source tools.

> **"GitHub Advanced Security and GitHub Actions enable us to run standardized scans across repositories throughout the enterprise. It gives us the best chance to stay as secure as possible when writing software."**
>
> *Senior principal engineer, technology*

Results. To account for these risks, Forrester adjusted this benefit downward by 5%, yielding a three-year, risk-adjusted total PV (discounted at 10%) of $5.2 million.

Efficiencies From Fewer Code Defects And Vulnerabilities

Ref.	Metric	Source	Year 1	Year 2	Year 3
C1	Number of software bugs and vulnerabilities reduced (annually)	Composite	6,870	7,370	7,870
C2	MTTR of code defects (hours)	Assumption	8	8	8
C3	Hourly fully loaded cost of a developer FTE	TEI standard	$75	$75	$75
C4	Productivity realization factor	TEI standard	50%	50%	50%
Ct	Efficiencies from fewer code defects and vulnerabilities	C1*C2*C3*C4	$2,061,000	$2,211,000	$2,361,000
	Risk adjustment	↓5%			
Ctr	Efficiencies from fewer code defects and vulnerabilities (risk-adjusted)		$1,957,950	$2,100,450	$2,242,950
	Three-year total: $6,301,350		Three-year present value: $5,201,025		

REDUCED DEVELOPER ONBOARDING TRAINING TIME THROUGH AUTOMATION

Evidence and data. Training a new developer is time-consuming. Interviewees said that before using GitHub, not only did their organization's new developers need to be trained on the organization's methods of coding, but developers also had to learn how to use the various tools that they would need during the SDLC. During that time, the interviewees' organizations were paying for unproductive work.

After implementing GitHub, these organizations drastically reduced the time to train new developers. This change was the result of a number of reasons:

- Many new hires graduate from college with a basic understanding of GitHub, and in environments that standardized on GitHub Enterprise Cloud, these new developers had fewer new tools to learn.
- GitHub Actions (through automated workflows) and Codespaces (through templates) eliminated mundane coding tasks, further reducing the number of things new developers needed to learn. They were able to contribute within days.
- Since new developers typically generate more code defects, GitHub Advanced Security and Dependabot proved helpful because they helped to catch coding errors and 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 from Day 1.

> **"Onboarding time from joining to being able to start committing code changes has gone down to being the first week of someone joining versus previously, when the first month of an engineer joining would have been written off as nonproductive for that engineer."**
>
> *Engineering director, financial consulting*

Modeling and assumptions. Based on the GitHub customer interviews, Forrester estimates the following about the composite organization:

- Due to turnover, the composite organization hires 350 new developers in Year 1, 750 in Year 2, and 850 in Year 3.
- The fully loaded hourly cost of a developer FTE is $75.
- Each new developer previously required ten days (80 hours) of training.
- Developer training time is reduced from 10 days to two days (16 hours), which is a reduction of 64 hours per new hire.
- The composite organization sees a 50% productivity recapture rate.

Risks. The benefit can vary by organization due to the following factors:

- The skill level of the new-hire developers.
- The complexity of the development environment.
- The effective implementation of automated workflows in GitHub.

Results. To account for these risks, Forrester adjusted this benefit downward by 15%, yielding a three-year, risk-adjusted total PV (discounted at 10%) of more than $3.2 million.

Reduced Developer Onboarding Training Time Through Automation

Ref.	Metric	Source	Year 1	Year 2	Year 3
D1	Number of developer new hires	Composite	350	750	850
D2	Hourly fully loaded cost of a developer FTE	TEI standard	$75	$75	$75
D3	New-hire training hours saved through simplified processes	Interviews	64	64	64
D4	Productivity realization factor	TEI standard	50%	50%	50%
Dt	Reduced developer onboarding training time through automation	D1*D2*D3*D4	$840,000	$1,800,000	$2,040,000
	Risk adjustment	↓15%			
Dtr	Reduced developer onboarding training time through automation (risk-adjusted)		$714,000	$1,530,000	$1,734,000
	Three-year total: $3,978,000		**Three-year present value: $3,216,334**		

IMPROVED DEVOPS AND SITE RELIABILITY ENGINEER EFFICIENCY

Evidence and data. Interviewed representatives noted that their organization's DevOps and DevSecOps teams previously spent a large portion of their time writing custom code to detect code defects and security vulnerabilities across all platforms. With GitHub Actions, they could create workflows that automated many of their tasks. With GitHub Codespaces, they could create templates, so every developer used the same approved codebase.

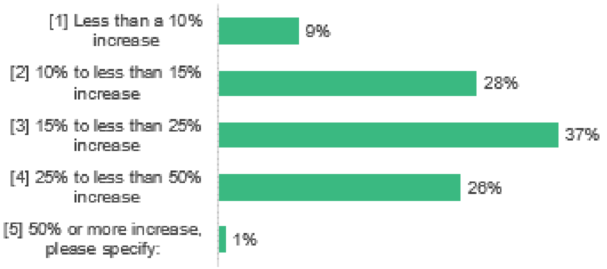
GitHub Actions, Codespaces, and Advanced Security reduced the time DevOps and DevSecOps teams spent managing the centralized developer environment by creating templates and workflows. The standardization and automation also reduced the number of software defects and security flaws DevOps and DevSecOps needed to remediate.

Additionally, several interviewees reported that by standardizing GitHub Enterprise Cloud, their organization could eliminate most of the centralized servers dedicated to other software tools because GitHub Enterprise Cloud performed those tools' functions in the cloud. This freed the DevOps and DevSecOps teams from operations management of hundreds of servers.

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

Survey participants noted the following productivity improvements:

"By what percentage did GitHub increase DevSecOps productivity for your organization?"



Base: 115 – GitHub Enterprise Cloud users who said it increased DevSecOps productivity at their organization
Source: A commissioned study conducted by Forrester Consulting on behalf of GitHub, August 2022

Modeling and assumptions. Based on the GitHub customer interviews, Forrester estimates the following about the composite organization:

- The composite organization employs 80 DevOps FTEs and 50 DevSecOps FTEs.
- The fully loaded hourly costs of a DevOps and DevSecOps FTE is $75.
- Of the 2,080 hours worked in a year, 75% of the time (1,560 hours) is spent managing operations, code quality, production rollouts, CI/CD, etc.
- DevOps and DevSecOps reduce the time spent on managing operations and remediating issues by 8% in Year 1, 12% in Year 2, and 15% in Year 3.
- The composite organization sees an 80% productivity recapture rate.

Risks. The benefit can vary by organization due to the following factors:

- The complexity of the development environment.
- The effective implementation of templates and automated workflows in GitHub.
- The number of legacy open source tools.

> **"We are working together with virtually thousands of other software suppliers, and every big supplier also has a stake in supplying software to us. We totally lost grasp of the source code, so one of the driving forces and main benefits of GitHub is that we can keep control of our source code."**
>
> *Deputy lead of dynamic platforms, automotive manufacturing*

Results. To account for these risks, Forrester adjusted this benefit downward by 10%, yielding a three-year, risk-adjusted total PV (discounted at 10%) of over $3.1 million.

Improved DevOps And Site Reliability Engineer Efficiency

Ref.	Metric	Source	Year 1	Year 2	Year 3
E1	Number of DevOps FTEs	Composite	80	80	80
E2	Number of DevSecOps FTEs	Composite	50	50	50
E3	Hourly fully loaded cost of a DevSecOps FTE	Composite	$75	$75	$75
E4	Hours worked on code quality, production rollout, CI/CD	Composite	1,560	1,560	1,560
E5	Reduced CI/CD, QA, and operations effort for DevOps and DevSecOps teams	Findings	8%	12%	15%
E6	Productivity realization factor	TEI standard	80%	80%	80%
Et	Improved DevOps and site reliability engineer efficiency	(E1+E2)*E3*E4*E5*E6	$973,440	$1,460,160	$1,825,200
	Risk adjustment	↓10%			
Etr	Improved DevOps and site reliability engineer efficiency (risk-adjusted)		$876,096	$1,314,144	$1,642,680
	Three-year total: $3,832,920		Three-year present value: $3,116,690		

TIME SAVINGS ON IT AUDIT PREPARATION

Evidence and data. Interviewees said that before using GitHub, their organizations had teams of auditors that would gather documentation from the decentralized environments. The nonstandard way of performing SDLC tasks led to an effort to merge the documentation from multiple repositories into a single format that could be used for audit purposes. In many cases, dedicated auditor teams worked full-time, manually performing internal and external audits.

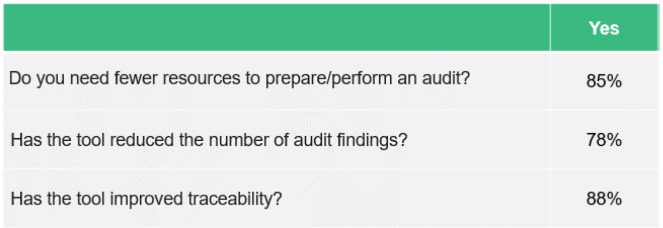
GitHub Enterprise Cloud automated documentation processes, reducing developers' time manually documenting code. These new, standardized documentation structures made it easier for auditors to find and compile the documentation necessary to be compliant. This helped the interviewees' organizations save time preparing for industry compliance and security audits.

An engineering director at a financial consulting firm reported that their organization implemented GitHub Pages to create documentation sites and GitHub Actions to automate the generation of crucial information. They explained: "We need to share certain procedures, standards, and many other things. GitHub Pages help us share that information with our teams and team members. I've used this documentation to answer our audits. Some of our audits ask us for organizational charts, roles and responsibilities, and things like that. [It's] documentation that lives on those GitHub Pages, and I've used that as evidence in audits."

> **"We consider source code an important asset and part of our intellectual property. So, one big driver to consider GitHub was our strategy to bring all source code together to have everything documented at one central location and to also have it under our control."**
>
> *Deputy lead of dynamic platform, automotive manufacturing*

Survey participants noted the following productivity improvements:

"You indicated that GitHub reduced the time to prepare/perform an audit for your organization. As a result of GitHub, how would you answer the following questions?"



	Yes
Do you need fewer resources to prepare/perform an audit?	85%
Has the tool reduced the number of audit findings?	78%
Has the tool improved traceability?	88%

Base: 117 – GitHub Enterprise Cloud users who said it reduced time to prepare/perform an audit at their organization
Source: A commissioned study conducted by Forrester Consulting on behalf of GitHub, August 2022

Modeling and assumptions. Based on the GitHub customer interviews, Forrester estimates the following about the composite organization:

- The composite organization employs 35 full-time auditors. Each auditor works 8 hours daily and 250 days per year preparing for audits.
- After implementing GitHub Enterprise Cloud, the auditors save 4 hours per day because the creation of the documentation is automated, and the format is standard. Using these tools cuts in half the hours auditors spend gathering information.
- The fully loaded cost for an auditor FTE is $45 per hour.
- The composite organization sees an 80% productivity recapture rate.

Risks. The benefit can vary by organization due to the following factors:

- The number of auditors and the average hourly cost of audit FTEs.
- The types of audits performed.

Results. To account for these risks, Forrester adjusted this benefit downward by 15%, yielding a three-year, risk-adjusted total PV (discounted at 15%) of over $2.6 million.

Time Savings On IT Audit Preparation

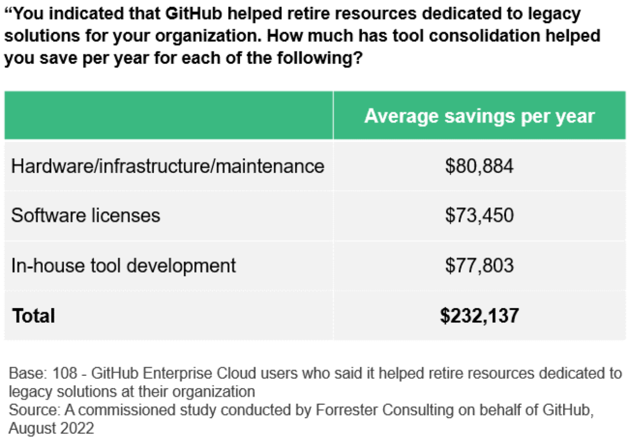
Ref.	Metric	Source	Year 1	Year 2	Year 3
F1	Number of auditors	Composite	35	35	35
F2	Number of hours saved (daily) from access to code documentation	Interviews	4	4	4
F3	Hourly fully loaded cost of an audit FTE	Composite	$45	$45	$45
F4	Productivity realization factor	TEI standard	80%	80%	80%
Ft	Time savings on IT audit preparation	F1*F2*F3*250 days*F4	$1,260,000	$1,260,000	$1,260,000
	Risk adjustment	↓15%			
Ftr	Time savings on IT audit preparation (risk-adjusted)		$1,071,000	$1,071,000	$1,071,000
	Three-year total: $3,213,000		**Three-year present value: $2,663,418**		

SAVINGS FROM RETIRING LEGACY DEVELOPMENT TOOLS

Evidence and data. The DevOps and DevSecOps teams at the interviewees' organizations managed centralized infrastructures for many legacy tools. By standardizing GitHub Enterprise Cloud, many of these servers were no longer needed and could be decommissioned over time.

GitHub Enterprise Cloud, Actions, and Advanced Security performed most of the centralized tasks legacy open source tools previously performed. Standardizing GitHub Enterprise Cloud and Advanced Security allowed the organizations to gradually retire centralized legacy tools and the hardware that supported them. The engineering director at the financial consulting firm said: "We're in the process of retiring other solutions. And, yes, we still use some other code-scanning systems for some very specific use cases and requirements, but they will be deprecated in favor of Advanced Security."

Survey participants noted the following productivity improvements:



"You indicated that GitHub helped retire resources dedicated to legacy solutions for your organization. How much has tool consolidation helped you save per year for each of the following?

	Average savings per year
Hardware/infrastructure/maintenance	$80,884
Software licenses	$73,450
In-house tool development	$77,803
Total	**$232,137**

Base: 108 - GitHub Enterprise Cloud users who said it helped retire resources dedicated to legacy solutions at their organization
Source: A commissioned study conducted by Forrester Consulting on behalf of GitHub, August 2022

Modeling and assumptions. Based on the GitHub customer interviews, Forrester estimates the following about the composite organization:

- The composite organization previously maintained a number of legacy tools and incurred annual infrastructure, software, and labor costs to sustain and manage these centralized environments.

- After implementing GitHub Enterprise Cloud, the composite organization gradually retires its legacy tools, resulting in accumulative savings of $250,000 in Year 1, $350,000 in Year 2, and $450,000 in Year 3.

Risks. The benefit can vary by organization due to the following factors:

- The size and nature of the organization's legacy software stack. Some tools will not be replaced if they perform tasks better suited for the organization.
- The number of legacy tools the organization can retire every year.

> **"I would say tens if not hundreds of build servers eliminated. I can easily say that."**
>
> *Senior principal engineer, technology*

Results. To account for these risks, Forrester adjusted this benefit downward by 5%, yielding a three-year, risk-adjusted total PV (discounted at 5%) of approximately $811,900.

Savings From Retiring Legacy Development Tools

Ref.	Metric	Source	Year 1	Year 2	Year 3
G1	Cost of legacy tools replaced by GitHub (e.g., infrastructure, software, and labor)	Findings	$250,000	$350,000	$450,000
Gt	Savings from retiring legacy development tools	G1	$250,000	$350,000	$450,000
	Risk adjustment	↓5%			
Gtr	Savings from retiring legacy development tools (risk-adjusted)		$237,500	$332,500	$427,500
	Three-year total: $997,500	Three-year present value: $811,890			

UNQUANTIFIED BENEFITS

Interviewees and survey respondents mentioned the following additional benefits that their organizations experienced but were not able to quantify:

Employee satisfaction. Skilled developers are in high demand. Employee retention is a constant battle. Offboarding and onboarding developers is expensive. Firms will do whatever is in their power to retain developers. Using GitHub saved the organizations' staff time and energy and allowed them to spend more time on productive activities. The more efficient the SDLC, the less stress on the developers. This increases employee satisfaction and retention of highly skilled resources in several IT departments**.** The director of architecture at the telecom firm noted: "There's a real benefit in that you can safely walk away to get a coffee for 5 minutes and the pull request is done when you come back as opposed to having to watch it because it scales very high The benefit is not necessarily direct productivity here; its' about trust, reliability, and predictability. That's a huge, huge win."

- **Customer satisfaction.** With fewer defects in their software code, the organizations saw less application downtime, while faster MTTR for problems that still make it into production increased application uptime and overall customer satisfaction. The deputy lead of dynamic platforms at the automotive manufacturing firm stated: "If you get a call from a customer, you want to be as fast as possible. We were definitely able to decrease the rate to the end customer to below 24 hours. So, that was pretty impressive on that side."

- **Better collaboration between teams.** GitHub enabled development teams to centralize enterprise code bases, making it easier for developers in different departments to share best practices with the organization. The engineering director at the financial consulting firm stated: "What GitHub enables us to do is create a 'parts warehouse' where developers can find useful code that can be quickly assembled into new solutions." The organizations experienced enhanced collaboration among developers across regions and teams. Security teams collaborate by using tools like GitHub Actions to perform secret scanning across the enterprise to uncover potential security flaws hidden in code.

FLEXIBILITY

The value of flexibility is unique to each customer. There are multiple scenarios in which a customer might implement GitHub and later realize additional uses and business opportunities, including:

Better code quality with Copilot. Some interviewees' organizations are experimenting with Copilot, an AI-driven tool that assists developers in writing better code by learning from past code and providing real-time suggestions. The engineering director at the financial consulting firm said: "We are really excited about Copilot as well, particularly the enterprise version of Copilot. It is able to learn from code that you have within your own enterprise, and it makes your engineers more effective."

Flexibility would also be quantified when evaluated as part of a specific project (described in more detail in Appendix A).

> **“Before GitHub, it was difficult to know how much code we had, how many repositories we had, and how much activity we had from developers. And many security vulnerabilities, like the Log4j, were still hidden in the code.”**
>
> Deputy lead of dynamic platforms - automotive manufacturing

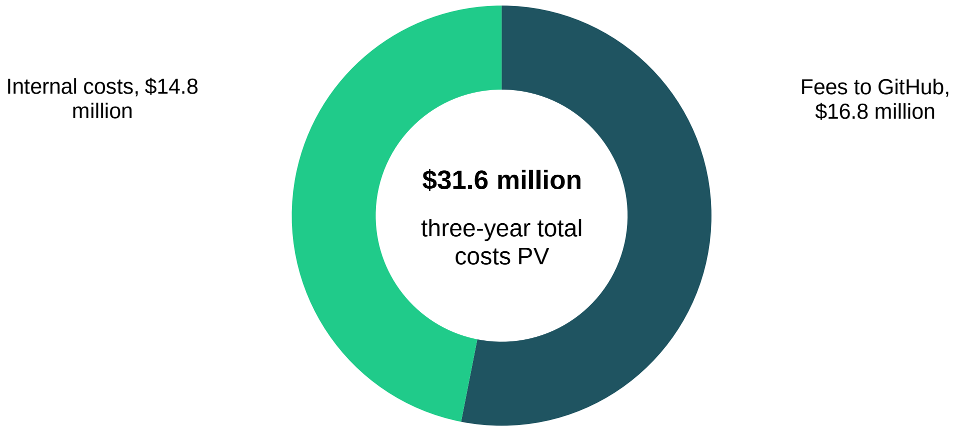
Analysis Of Costs

🟩 Quantified cost data as applied to the composite

Total Costs

Ref.	Cost	Initial	Year 1	Year 2	Year 3	Total	Present Value
Htr	Fees to GitHub	$50,400	$6,312,600	$6,759,900	$7,207,200	$20,330,100	$16,790,698
Itr	Internal Costs	$8,919,900	$1,634,325	$2,748,075	$2,871,825	$16,174,125	$14,834,431
	Total costs (risk-adjusted)	$8,970,300	$7,946,925	$9,507,975	$10,079,025	$36,504,225	$31,625,129

COSTS BY CATEGORY



This section examines the costs incurred with licensing, setting up, customizing, and managing the GitHub platform.

FEES TO GITHUB

Evidence and data. This cost represents external costs interviewees' organizations pay directly to GitHub. GitHub operates on a subscription model and charges licensing fees per user. In addition, organizations typically employ GitHub consultants to assist in the initial setup of the environment, the initial development of code, and the sharing of best practices.

Modeling and assumptions. Forrester estimates the following about the composite organization:

- The composite organization subscribes to GitHub Enterprise Cloud with a licensing fee of $21 per user per month and Advanced Security with a fee of $49 per user per month.
- The composite organization requires licenses for 7,000 developers in Year 1, growing to 7,500 in Year 2 and to 8,000 in Year 3.
- GitHub Actions and Codespaces usage fees are based on the compute minutes a process is active. It will automatically stop billing after 30 minutes of inactivity. Annual usage fees for the composite organization increase year-over-year based on the usage by a growing number of developers.
- The composite organization employs a GitHub consultant to assist in implementing the GitHub solution stack. The consultant is hired for 160 hours at $300 per hour.

Risks. The risk can vary by organization due to the following factors:

- The number of users and GitHub Actions and Codespaces usage requirements.
- The effort required to stand up the GitHub environment.

Results. To account for these risks, Forrester adjusted this cost upward by 5%, yielding a three-year, risk-adjusted total PV (discounted at 5%) of $16.8 million.

Fees To GitHub

Ref.	Metric	Source	Initial	Year 1	Year 2	Year 3
H1	Number of Developer FTEs	Composite		7,000	7,500	8,000
H2	GitHub Enterprise Cloud license fees (monthly)	GitHub		$21	$21	$21
H3	GitHub Advanced Security license fees (monthly)	GitHub		$49	$49	$49
H4	Subtotal: GitHub Enterprise Cloud and Advanced Security license fees	H1*(H2+H3) * 12 months		$5,880,000	$6,300,000	$6,720,000
H5	**GitHub Actions and Codespaces usage fees**	**Interviews**		**$84,000**	**$90,000**	**$96,000**
H6	GitHub consultant hourly rate	Interviews	$300	$300	$300	$300
H7	Number of consulting hours (annually)	Composite	160	160	160	160
H8	**GitHub consulting services fees**	**H6*H7**	**$48,000**	**$48,000**	**$48,000**	**$48,000**
Ht	Fees to GitHub	H4+H5+H8	$48,000	$6,012,000	$6,438,000	$6,864,000
	Risk adjustment	↑5%				
Htr	Fees to GitHub (risk-adjusted)		$50,400	$6,312,600	$6,759,900	$7,207,200
	Three-year total: $20,330,100			**Three-year present value: $16,790,698**		

INTERNAL COSTS

Evidence and data. Internal costs represent those the interviewees' organizations incurred to implement and manage GitHub and train employees on GitHub.

Modeling and assumptions. Forrester estimates the following about the composite organization:

- The composite organization dedicates seven DevOps and DevSecOps FTEs to managing the centralized environment. Initially, six FTEs work with GitHub consultants for three months to install and configure the software and to write management code.
- Each developer requires initial training on GitHub and the new standards and protocols. Initially, 7,000 developers receive 15 hours of training.
- In Years 2 and 3, an additional 500 developers are onboarded into GitHub.
- Due to attrition, additional new hires require training in Years 1, 2, and 3.
- The fully loaded cost for a developer FTE, DevOps FTE, and DevSecOps FTE is $75 per hour.

Risks. The risk can vary by organization due to the following factors:

- The number of FTEs required to perform the initial installation and the ongoing operational work, which may vary depending on the size and complexity of the organization.
- Training requirements may vary depending on the coding requirements.

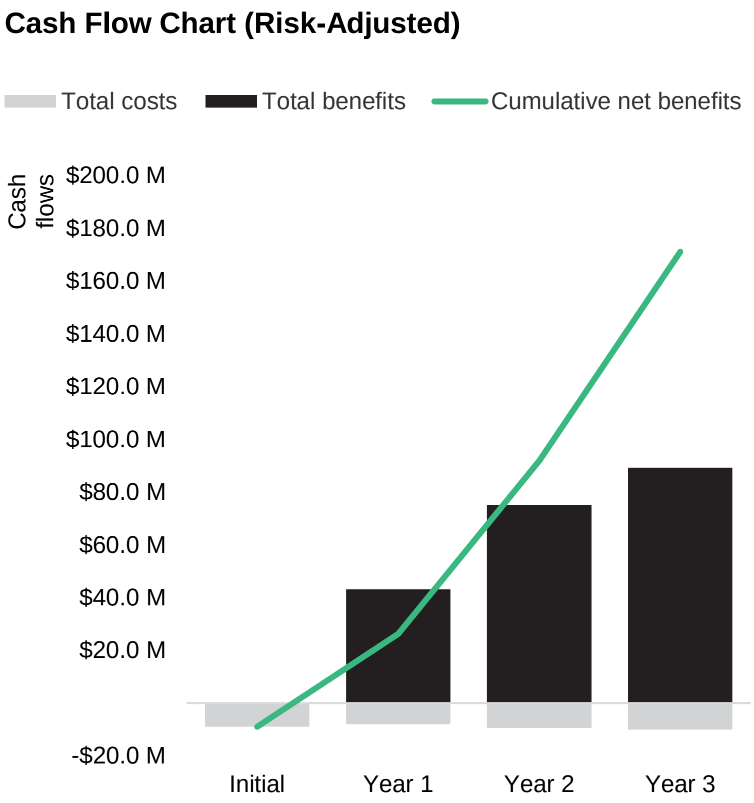
Results. To account for these risks, Forrester adjusted this cost upward by 10%, yielding a three-year, risk-adjusted total PV (discounted at 10%) of $14.8 million.

Internal Costs

Ref.	Metric	Source	Initial	Year 1	Year 2	Year 3
I1	Number of DevOps FTEs	Composite	6	7	7	7
I2	Number of hours customizing GitHub	Composite	520	2,080	2,080	2,080
I3	Hourly fully loaded cost of a DevOps FTE	TEI standard	$75	$75	$75	$75
I4	Subtotal: Cost of Customizing GitHub	I1*I2*I3	$234,000	$1,092,000	$1,092,000	$1,092,000
I5	Number of developers to be trained	Composite	7,000	0	500	500
I6	Number of new hires to be trained	Composite	0	350	750	850
I7	Hourly fully loaded cost of a FTE	TEI standard	$75	$75	$75	$75
I8	Number of training hours	Composite	15	15	15	15
I9	Subtotal: Developer training	(I5+I6)*I7*I8	$7,875,000	$393,750	$1,406,250	$1,518,750
It	Internal costs	I4+I9	$8,109,000	$1,485,750	$2,498,250	$2,610,750
	Risk adjustment	↑10%				
Itr	Internal costs (risk-adjusted)		$8,919,900	$1,634,325	$2,748,075	$2,871,825
	Three-year total: $16,174,125			**Three-year present value: $14,834,431**		

Financial Summary

CONSOLIDATED THREE-YEAR RISK-ADJUSTED METRICS



The financial results calculated in the Benefits and Costs sections can be used to determine the ROI, NPV, and payback period for the composite organization's investment. Forrester assumes a yearly discount rate of 10% for this analysis.

These risk-adjusted ROI, NPV, and payback period values are determined by applying risk-adjustment factors to the unadjusted results in each Benefit and Cost section.

Cash Flow Analysis (Risk-Adjusted Estimates)

	Initial	Year 1	Year 2	Year 3	Total	Present Value
Total costs	($8,970,300)	($7,946,925)	($9,507,975)	($10,079,025)	($36,504,225)	($31,625,129)
Total benefits	$0	$43,106,902	$75,223,508	$89,264,108	$207,594,519	$168,421,727
Net benefits	($8,970,300)	$35,159,977	$65,715,533	$79,185,083	$171,090,294	$136,796,598
ROI						433%
Payback						<6 months

Appendix A: Total Economic Impact

Total Economic Impact is a methodology developed by Forrester Research that enhances a company's technology decision-making processes and assists vendors in communicating the value proposition of their products and services to clients. The TEI methodology helps companies demonstrate, justify, and realize the tangible value of IT initiatives to both senior management and other key business stakeholders.

TOTAL ECONOMIC IMPACT APPROACH

Benefits represent the value delivered to the business by the product. The TEI methodology places equal weight on the measure of benefits and the measure of costs, allowing for a full examination of the effect of the technology on the entire organization.

Costs consider all expenses necessary to deliver the proposed value, or benefits, of the product. The cost category within TEI captures incremental costs over the existing environment for ongoing costs associated with the solution.

Flexibility represents the strategic value that can be obtained for some future additional investment building on top of the initial investment already made. Having the ability to capture that benefit has a PV that can be estimated.

Risks measure the uncertainty of benefit and cost estimates given: 1) the likelihood that estimates will meet original projections and 2) the likelihood that estimates will be tracked over time. TEI risk factors are based on "triangular distribution."

The initial investment column contains costs incurred at “time 0” or at the beginning of Year 1 that are not discounted. All other cash flows are discounted using the discount rate at the end of the year. PV calculations are calculated for each total cost and benefit estimate. NPV calculations in the summary tables are the sum of the initial investment and the discounted cash flows in each year. Sums and present value calculations of the Total Benefits, Total Costs, and Cash Flow tables may not exactly add up, as some rounding may occur.

PRESENT VALUE (PV)

The present or current value of (discounted) cost and benefit estimates given at an interest rate (the discount rate). The PV of costs and benefits feed into the total NPV of cash flows.

NET PRESENT VALUE (NPV)

The present or current value of (discounted) future net cash flows given an interest rate (the discount rate). A positive project NPV normally indicates that the investment should be made, unless other projects have higher NPVs.

RETURN ON INVESTMENT (ROI)

A project’s expected return in percentage terms. ROI is calculated by dividing net benefits (benefits less costs) by costs.

DISCOUNT RATE

The interest rate used in cash flow analysis to take into account the time value of money. Organizations typically use discount rates between 8% and 16%.

PAYBACK PERIOD

The breakeven point for an investment. This is the point in time at which net benefits (benefits minus costs) equal initial investment or cost.

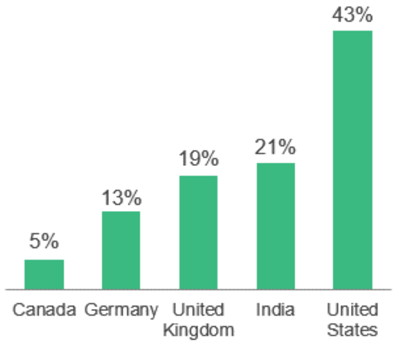
Appendix B: Interview And Survey Demographics

Interviews

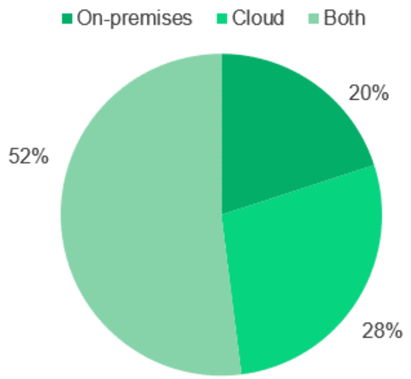
Role	Industry	Revenue	Number of GitHub Users
Senior principal engineer	Technology	$1 billion	2,000 globally
Director of architecture	Telecom	$17 billion	5,000 in North America
Engineering director	Financial consulting	$32 billion	5,200 globally
Deputy lead of dynamic platforms	Automotive manufacturing	$130 billion	22,000 globally (internal and external)

Survey Demographics

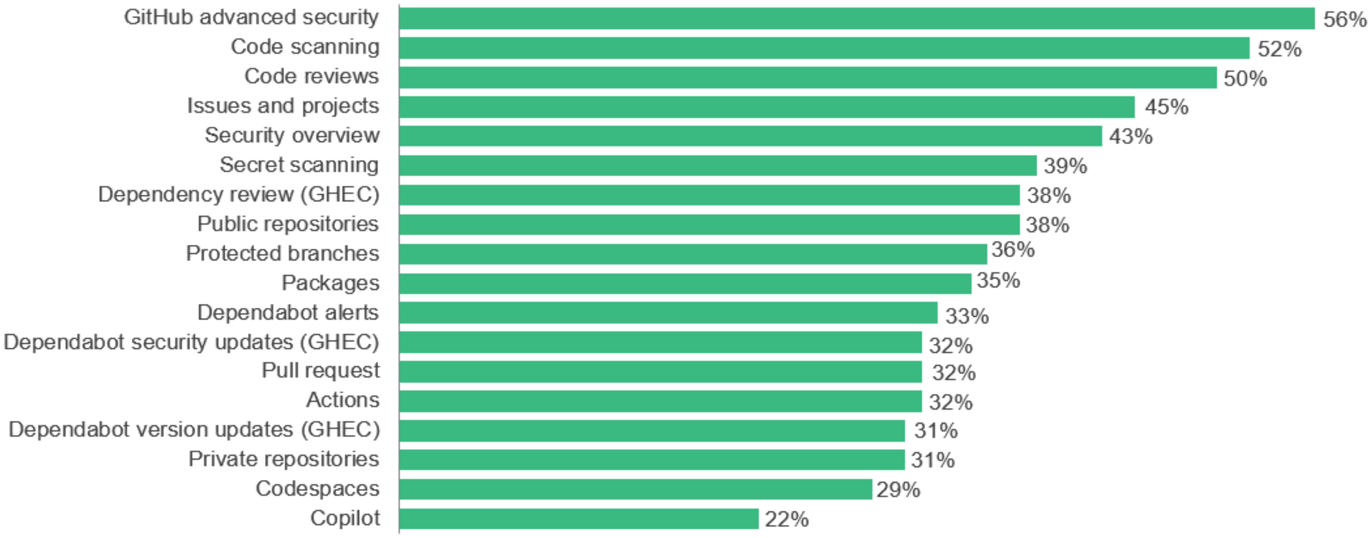
"In which country are you located?"



"How are you currently deploying GitHub?"



"What features of GitHub are you using?"



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

Appendix C: Endnotes

[1] Total Economic Impact is a methodology developed by Forrester Research that enhances a company's technology decision-making processes and assists vendors in communicating the value proposition of their products and services to clients. The TEI methodology helps companies demonstrate, justify, and realize the tangible value of IT initiatives to both senior management and other key business stakeholders.

FORRESTER®