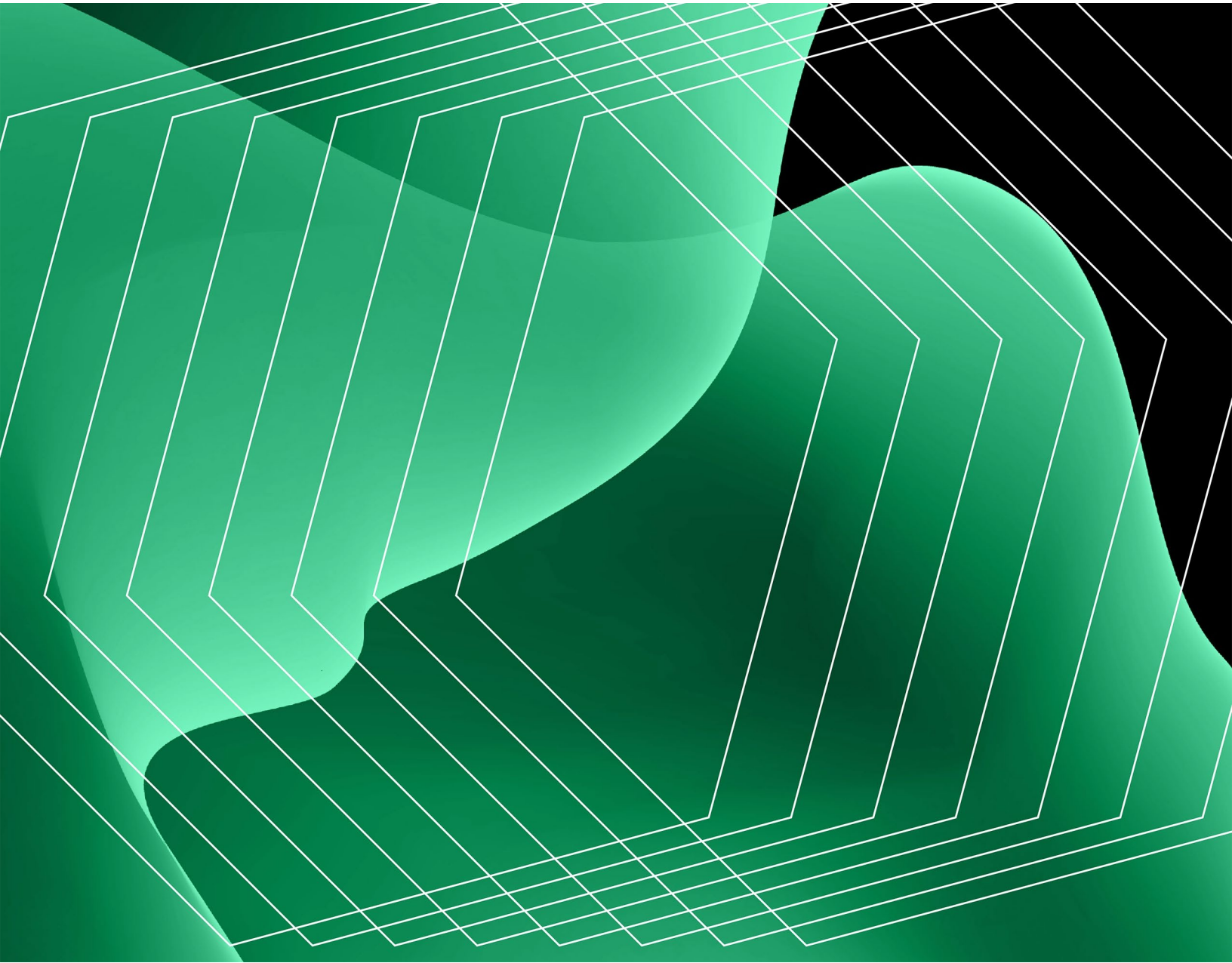
FORRESTER®

# The Total Economic Impact™ Of GitLab Ultimate

Cost Savings And Business Benefits Enabled By GitLab Ultimate

A Forrester Total Economic Impact™ Study
Commissioned By GitLab, October 2024



# Table Of Contents




Consulting Team:

Jeffrey Yozwiak

Zahra Azzaoui


**ABOUT FORRESTER CONSULTING**

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

# Executive Summary

**Organizations with fragmented toolchains risk slow software delivery, higher IT costs, and higher security risks — issues that only keep growing as tool sprawl increases. As an integrated software delivery platform (IDSP), GitLab is a single solution for all stages of the development, security, and operations (DevSecOps) lifecycle. Organizations that replaced point solutions and consolidated their toolchains with GitLab improved developer productivity and happiness, lowered their IT costs, and enhanced security. Delivering better software faster — while maintaining the highest security and quality standards — ultimately drove business growth.**

GitLab is a comprehensive DevSecOps platform supporting every stage of the software development lifecycle (SDLC) — from initial planning all the way through to production delivery and monitoring and analytics. GitLab is an alternative to fragmented software delivery toolchains comprised of numerous point solutions. Organizations can use GitLab to meet all their software development tooling needs, or they can use GitLab in combination with other tools; organizations with legacy DevSecOps toolchains often adopt GitLab incrementally, replacing their prior toolsets over time. As a single, unified platform, GitLab can improve developer productivity by reducing context switching, streamlining workflows, and incorporating security into every step of the SDLC. With GitLab Duo, GitLab also integrates AI throughout SDLC, further supporting developer and team productivity. GitLab's enterprise offering is known as GitLab Ultimate.

GitLab commissioned Forrester Consulting to conduct a Total Economic Impact™ (TEI) study and examine the potential return on investment (ROI) enterprises may realize by deploying GitLab Ultimate.[1] The purpose of this study is to provide readers with a framework to evaluate the potential financial impact of GitLab Ultimate on their organizations.

Return on investment (ROI)

**483%**

Net present value (NPV)

**$74.6M**

To better understand the benefits, costs, and risks associated with this investment, Forrester interviewed eight representatives from four organizations using GitLab Ultimate. For the purposes of this study, Forrester aggregated the interviewees' experiences and combined the results into a single composite organization. This composite organization is a $5-billion company with 5,000 employees, of which there are 2,000 employees contributing to software delivery (e.g., developers, etc., on cross-functional agile teams). Half of the composite organization's annual revenue is driven by software development, and the composite operates in a business environment that makes software security and quality especially crucial.

Before investing in GitLab, the interviewees' organizations had fragmented, sprawling toolchains that were expensive both to license and to manage. Worse, the fragmented nature of the toolchains was hurting software delivery: workflows were cumbersome, collaboration was low, and timelines were long. Because development processes were so inefficient, maintaining high standards for software quality and security was time-consuming and high-effort.

However, investing in GitLab Ultimate enabled the interviewees' organizations to retire redundant tools and consolidate their toolchains around the GitLab platform. The IT teams realized direct savings by sunsetting software and standardizing tooling. Because GitLab supported more efficient, integrated, and automated workflows, developer productivity improved, and the software development teams delivered projects faster than before. Despite the increased velocity, though, security and quality remained as high as they were before. By integrating security throughout the SDLC, teams were able to catch and fix issues sooner. These improvements in software delivery were boons to business.

## KEY FINDINGS

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

- **Software development team productivity gains worth $39.9 million.** Each developer saves 535 hours per year as regular activities all become more efficient with GitLab. Testing automation enabled by GitLab not only saves developers time, but it also empowers them to test more frequently. Developers

detect issues sooner and debug them faster, which improves developer productivity because issues are identified earlier in the SDLC when they are easiest to fix. Overall, having a single, unified interface for all development tasks streamlines workflows, reduces context switching and overhead, and improves collaboration and the developer experience.

## 305 hours per developer per year

Time reclaimed from inefficient workflows and redirected towards coding[2]

- **Faster developer onboarding saves $2.3 million.** With GitLab, new hires to the composite organization's software development team ramp up to full productivity 75% faster (i.e., in weeks instead of months). GitLab improves knowledge transfer by helping to standardize workflows and processes, increase documentation, and foster collaboration.

## 1.5 months

Time to onboard a new developer before GitLab

## 1.5 weeks

Time to onboard a new developer after GitLab

- **Faster time to market for new software yield benefits of $37.7 million.** The productivity gains unlocked with GitLab enable the composite's software development team to deliver 50% more work with business value than it otherwise would have. Projects are completed faster and sooner than without

GitLab, driving revenue. Despite the increase in velocity, software quality and security remain at the same high levels, as the next benefit explains.

**50%**

More features delivered

- **Security efficiencies save $5.9 million.** With GitLab, the cybersecurity and software development teams at the composite organization continue to maintain issue-free software with 81% less effort. GitLab enables these teams to integrate security protocols and scanning throughout all stages of the SDLC (moving security from a final step before release to something every team member is responsible for). As collaboration between developers and security increases, overall issues decline and any potential issues are surfaced and fixed sooner. GitLab also improves visibility into software development processes, thereby reducing the time security and development team members must spend on recurring activities, such as disaster recovery prep and auditing and compliance.

**78 hours per security team member per year**

Time reclaimed and redirected towards enhancing security[3]

- **Toolchain consolidation — retiring unnecessary legacy tools — saves $4.3 million in licensing and administration costs.** GitLab makes many legacy tools redundant, and retiring tools reduces total spending on software licensing by 25%. The effort that IT spends supporting its toolchain also falls by 75%; the

retired legacy tools no longer need to be maintained and a simplified toolchain standardized around GitLab is easier for IT administrators to support.

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

- **Better developer experience.** Productive developers are happy developers. Using GitLab removes frustrations and fosters collaboration and a sense of community.
- **Superior support.** GitLab's partnership and commitment to users empowers organizations to focus on the business challenges of building software rather than on maintaining their toolchains and pipelines.
- **GitLab Duo AI.** GitLab Duo integrates AI throughout every stage of the development process. While these features are a nascent part of GitLab's platform, they have the potential to exponentially improve developer productivity and software quality.

> "The look and feel is fantastic. It allows us to do so many things we couldn't do before."
>
> SOFTWARE ARCHITECT, ENERGY/RESEARCH

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

- **Licensing costs of $3.4 million.** GitLab Ultimate's flexible licensing model is based on usage.
- **Implementation costs of $744,000.** Deploying GitLab throughout the composite organization involves effort from IT administrators as well as (optional) professional services.
- **Ongoing management costs of $1.1 million.** Ongoing administration of GitLab is relatively minimal.

- **On-premises infrastructure costs of $5.4 million.** Only organizations deploying GitLab on-premises like the composite organization incur this cost.
- **New user training costs $4.8 million.** New GitLab users spend about 40 hours on training. This training is flexible (e.g., introductory videos at first and advanced tutorials later on).

The representative interviews and financial analysis found that a composite organization experiences benefits of $90.0 million over three years versus costs of $15.4 million, adding up to a net present value (NPV) of $74.6 million and an ROI of 483%.

Return on investment (ROI)

**483%**

Benefits PV

**$90.0M**

Net present value (NPV)

**$74.6M**

Payback

**<6 months**

## Benefits (Three-Year)

## TEI FRAMEWORK AND METHODOLOGY

From the information provided in the interviews, Forrester constructed a Total Economic Impact™ framework for those organizations considering an investment in GitLab Ultimate.

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 GitLab Ultimate can have on an organization.

**DISCLOSURES**
Readers should be aware of the following:

This study is commissioned by GitLab 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 GitLab Ultimate.

GitLab 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.

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

**Due Diligence**
Interviewed GitLab stakeholders and Forrester analysts to gather data relative to GitLab Ultimate.

**Interviews**
Interviewed eight representatives from four organizations using GitLab Ultimate to obtain data about costs, benefits, and risks.

**Composite Organization**
Designed a composite organization based on characteristics of the interviewees' organizations.

**Financial Model Framework**
Constructed a financial model representative of the interviews using the TEI methodology and risk-adjusted the financial model based on issues and concerns of the interviewees.

**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.

# The GitLab Ultimate Customer Journey

Drivers leading to the GitLab Ultimate investment

| Interviews | | | | | |
|---|---|---|---|---|---|
| **Role** | **Industry** | **Revenue** | **Geography** | **Employees** | **GitLab Users** |
| IT manager (DevSecOps CoE)<br>Senior consultant (EA and DevSecOps)<br>Supervisor (service management technology) | Finance | $4B | North America | 3,000 | 150 |
| Program manager (FinDev platform) | Finance | $1.5B | EMEA | 5,000 | 2,000 |
| Software architect (cloud-native systems) | Energy/research | $1.3B | EMEA | 2,500 | 19,000 |
| CTO and SVP<br>Product owner (common software development environment)<br>Consultant (DevSecOps) | Defense | $7.5B | North America | 25,000 | 2,000 |

## KEY CHALLENGES

Before investing in GitLab Ultimate, the interviewees' organizations grappled with significant challenges due to fragmented delivery environments. They described a reliance on a collection of disjointed tools: version control, continuous integration and continuous delivery (CI/CD), code review, security testing, and project management were handled by separate systems that lacked integration. This fragmented setup created silos within teams, where collaboration was limited and workflows were inefficient. The interviewees' organizations also struggled with managing multiple tools, which led to high licensing and management costs. Interviewees also noted that their existing tools, while functional, were slow and cumbersome, making it difficult to maintain visibility, security, and compliance across the software development lifecycle.

Ultimately, these inefficiencies hindered their ability to meet business needs and remain competitive.

The interviewees noted how their organizations struggled with common challenges, including:

- **Fragmented tools led to inefficiencies and poor collaboration.** Before implementing GitLab Ultimate, the interviewees' organizations operated with toolchains that were disconnected and decentralized, resulting in a fragmented environment that disrupted smooth operations across software development and delivery teams. The use of disparate tools by different teams created isolated workspaces with limited interaction, which impeded collaboration and significantly slowed project progress.
- **High costs due to tool sprawl.** Managing and licensing a myriad of disparate tools burdened the interviewees' organizations with both direct licensing costs and administrative and support efforts required to keep them up to date and running smoothly.

  The CTO and SVP at the defense organization said: "One of the reasons we chose to invest in GitLab was to eliminate tool sprawl. As a large company that has acquired many organizations over time, we were burdened with a large toolset, which was highly inefficient. A significant percentage of our labor hours was spent on tool administration rather than on developing software and improving velocity."

> "Our superpower is software. It's measured in terms of velocity and the ability to get new capabilities into the hands of our customers. For that to remain our primary focus, it just made economic sense to [consolidate] onto a single platform."
>
> **CTO AND SVP, DEFENSE**

- **Limited visibility hindered compliance.** The interviewees' organizations lacked a centralized platform to track and monitor activities across various tools and departments, which complicated efforts to ensure adherence to regulatory standards and internal policies, hindered effective auditing, and ultimately increased the risk of noncompliance.

  The IT manager in finance said: "[Before GitLab], we were in the Wild West. We had code in a million different places. After we adopted GitLab, we started tracking metrics to see who is working on code quality scanning, auto deploys, automated builds, etc. We started reporting on these metrics every month and we saw a significant improvement in visibility. Now, we know who's working on what at a given time."

- **Significant manual effort extended development timelines.** Development cycles were often prolonged due to heavy reliance on manual processes, frequent context switching between tools, and cumbersome security practices. These inefficiencies led to significant delays that made it increasingly difficult to promptly meet business needs and remain competitive in the market.

- **Increased risk of security issues.** Complex and disconnected toolchains created quality and security risks. For the interviewees' organizations, software quality was crucial, but maintaining high quality and security standards was labor-intensive and time-consuming.

  For example, the program manager in finance explained: "We wanted to shift left because our security and quality checks were previously done after the build in an inefficient manner, using Excel spreadsheets that were sent via email to the team. Our teams had to deal with constant context switching and significant time lags. Every time we wanted to make a change to the pipeline, we had to ask the teams multiple times — twice, three times, or even four times. It was horrible."

## INVESTMENT OBJECTIVES

The interviewees' organizations searched for a solution that could:

- Consolidate multiple tools into a single, integrated platform.

- Enhance developer productivity.
- Ensure compliance with industry regulations and internal policies.
- Strengthen security throughout the software development lifecycle.

> "The menagerie of tools we had in our prior environment were old and not aligned with current software development practices. So, our decision to invest in GitLab was really a modernization effort. GitLab Ultimate provided our users with modern features and capabilities that they didn't have before."
>
> CTO AND SVP, DEFENSE

> "With GitLab, we have something in the cloud that everybody could use. Its scalable, secure, and flexible."
>
> CTO AND SVP, DEFENSE

**COMPOSITE ORGANIZATION**

Based on the interviews, 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 eight interviewees from four organizations, 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 regional, $5-billion company with 5,000 employees. Software development is crucial to the composite organization's business. In any given year, around half of the organization's revenue is attributable to its proprietary software (both software developed previously as well as new software enhanced or developed during the year). The other half of the organization's annual revenues come from work that is not directly tied to software development (e.g., consulting, research, physical product sales, etc.) Accordingly, around 40% of the composite organization's workforce is involved in software delivery: the organization has 2,000 employees (e.g., developers, etc.) working on cross-functional agile teams.

Security is also crucial for the composite organization, and it has an additional 65 employees on its security team. Security is crucial for business reasons. The organization develops software for critical infrastructure and essential systems. Any security or quality issues could result in significant business disruptions or regulatory fines and, in some cases, issues might even impact human safety. The composite organization therefore works to maintain exceptionally high quality standards for its software. All applications must be error-free, reliable, robust, and impenetrably secure. Security is important to the organization's brand, and it receives considerable attention from both leadership and customers. The security team is responsible for everything from application security to software supply chain security to cybersecurity and so on.

To keep this study as broadly applicable as possible, Forrester has not specified the composite organization's industry. However, there are multiple private-sector and public-sector industries in which organizations face these same software development challenges, such as high tech, finance, manufacturing, defense, etc.

**Deployment characteristics.** Before implementing GitLab Ultimate, the composite organization had a fragmented development environment and relied on a variety of disconnected tools. However, over three years, the organization strategically deploys GitLab Ultimate, focusing on areas such as security and compliance, CI/CD, automated software delivery, agile delivery, SCM, and GitOps. The composite organization first rolls out GitLab Ultimate to around 10% of its software development and security team members in Year 1. After this initial effort is successful, the composite organization

expands GitLab usage and rolls out the platform to 50% of its workforce in Year 2. By Year 3, the composite organization's migration to GitLab is complete and all software development and security team members are using the platform. As the organization migrates to GitLab, it retires much of its prior tooling (however, some legacy tools remain in use).

**Key Assumptions**

$5 billion in revenue

2,000 software development team members

65 security team members

Three-year deployment timeline

**Composite Organization's Migration To GitLab**

| Ref. | Metric | Source | Year 1 | Year 2 | Year 3 |
|---|---|---|---|---|---|
| R1 | Total employees | Composite | 5,000 | 5,000 | 5,000 |
| R2 | Total software development team members | Composite | 2,000 | 2,000 | 2,000 |
| R3 | Total security team members | Composite | 65 | 65 | 65 |
| R4 | Percentage of software development and security team members using GitLab | Interviews | 10% | 50% | 100% |
| R5 | Software development team members using GitLab | R2*R4 | 200 | 1,000 | 2,000 |
| R6 | Security team members using GitLab | R3*R4 | 7 | 33 | 65 |

# 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 | Software development team productivity | $3,176,654 | $15,883,269 | $31,766,539 | $50,826,461 | $39,881,206 |
| Btr | Improved productivity of new hires to the software development team | $184,360 | $921,797 | $1,843,594 | $2,949,751 | $2,314,535 |
| Ctr | Accelerated time to market | $2,999,880 | $14,999,400 | $29,998,800 | $47,998,080 | $37,661,904 |
| Dtr | Security efficiencies | $469,069 | $2,337,619 | $4,671,347 | $7,478,035 | $5,867,995 |
| Etr | Toolchain consolidation | $340,875 | $1,704,375 | $3,408,750 | $5,454,000 | $4,279,505 |
| | Total benefits (risk-adjusted) | $7,170,838 | $35,846,460 | $71,689,030 | $114,706,328 | $90,005,145 |

## SOFTWARE DEVELOPMENT TEAM PRODUCTIVITY

**Evidence and data.** Interviewees reported that GitLab empowered their software development teams to work more efficiently by reducing the time spent testing, identifying and remediating issues, and conducting other routine tasks — ultimately enabling teams to focus less on manual tasks across the CI/CD pipeline and more on innovation and delivering high-quality software.

Interviewees reported software development team productivity gains for the following activities:

| Software Development Productivity Gains Reported By Interviewees | | |
|---|---|---|
| **Activity** | **How GitLab Helped** | **Example Successes** |
| Testing | GitLab's integrated CI/CD pipelines allowed developers to run tests continuously without manual intervention. This automation reduced the time developers spent on testing activities, freeing them to focus on higher-value tasks and accelerating the software development lifecycle while maintaining high-quality code. | • Each developer saved 20 hours per month due to GitLab's automated testing.<br>• Automated testing processes that previously took several hours were now 13 times faster with GitLab. |
| Debugging | With features like real-time monitoring, automated alerts, and integrated debugging tools, developers were able to quickly pinpoint and resolve problems earlier in the software development lifecycle, when they are easier to fix (this is known as "shifting left"). This rapid feedback loop minimized downtime and prevented issues from escalating and prolonging development. | • Time to fix faulty code and rerun tests was reduced from multiple days to a couple of hours.<br>• Issues (e.g., file system errors) previously took over an hour per error to resolve but were fixed in minutes with GitLab. |
| Miscellaneous nonproductive tasks | By unifying development, testing, and deployment workflows into a single interface, GitLab reduced context switching, enabling developers to stay focused without having to switch between multiple tools and environments. This approach allowed software development teams to complete routine tasks more quickly and streamlined the entire development process. | • Reduced context switching with GitLab saved each developer 20 minutes per activity. |

Interviewees reported different improvements in software development team productivity depending on their organizations' operational frameworks and development practices. These improvements varied across teams, but interviewees consistently highlighted the impact of GitLab Ultimate in streamlining workflows and reducing inefficiencies.

> "We've not only increased developer productivity — we've also improved the overall quality and security of our software."
>
> **SOFTWARE ARCHITECT, ENERGY/RESEARCH**

The interviewees described their experiences with GitLab as follows:

- The program manager in finance explained: “We initially estimated how much time we would save with GitLab and decided to be conservative by assuming we’d only achieve half of the time savings we expected to see. We then calculated the potential time savings and adjusted our teams accordingly. … In the end, the teams using GitLab were able to deliver the same results with fewer [resources (e.g., time)].”
- The software architect in energy/research said: “Before, if files needed to be placed in specific file systems and someone made a mistake, identifying who made the error and fixing it could take over an hour due to lack of version control. Now, we can pinpoint those issues in under a minute — what once took over an hour can now be resolved almost instantly.”

  They concluded: “When I have conversations about GitLab with our developers, they universally agree that it has increased productivity at our organization across teams and roles. We now have one platform that has functions that everyone can use.”

> “[Developers] had dealt with a lot of context switching between multiple tools for project management. [But now], everything is integrated in one place. You can create an issue, link it to a merge request, and close it all within the same platform. … Now, our developers no longer deal with context switching. With the time they’re saving, they’re working on more important matters.”
>
> **SOFTWARE ARCHITECT, ENERGY/RESEARCH**

> "[Before GitLab], our teams had to gather open-source tools, build stacks, and write scripts for a program to start up. Now, it's instant. It was like switching a light switch. We went from doing so much prework to doing none at all. It was an on-off switch."
>
> CTO AND SVP, DEFENSE

**Modeling and assumptions.** Based on the interviews, Forrester assumes that software development team members at the composite organization save time on their regular testing, debugging, and other miscellaneous tasks with GitLab. For each activity, Forrester makes the following specific assumptions for the composite organization:

- **For testing time avoided with GitLab:**
  - Before GitLab, software development team members spend about 13.5% of their annual work time testing software.[4]
  - With GitLab, the amount of time software development team members spend testing annually falls by 92% (due to testing automation, etc.).
- **For debugging time avoided with GitLab:**
  - Before GitLab, software development team members spend about 11.8% of their annual work time debugging software (i.e., tracking down, identifying, and resolving issues).[5]
  - With GitLab, the amount of time software development team members spend debugging falls by 80% (due to shifting left, etc.).
- **For miscellaneous activity time avoided with GitLab:**
  - Before GitLab, software development team members spend 7.5% of their annual work time on miscellaneous activities unrelated to programming.[6]

- With GitLab, the amount of time software development team members lose on unproductive miscellaneous activities falls by 50% (due to reduced context-switching, etc.).

Forrester also assumes:

- The average fully burdened annual salary for a software development team member is $138,000.[7] Also, team members typically work 40 hours per week (2,080 hours per year).
- Software development team members capture and use 50% of the time they save for productive activities that generate value for the business. The software development team members naturally use some of their time savings on activities that do not directly generate value for the business (e.g., coffee breaks, chats with coworkers, etc.).

| Time Savings Per GitLab User | | |
|---|---|---|
| **Activity** | **Time Per Activity Annually Before Gitlab** | **Time Per Activity Annually After GitLab** |
| Testing | 280.8 hours | 22.5 hours |
| Debugging | 257.0 hours | 58.4 hours |
| Miscellaneous nonproductive tasks | 156.0 hours | 78.0 hours |
| **Total** | **693.8 hours** | **158.9 hours** |

**Risks.** The risk (or likelihood) of this benefit varying from organization to organization is relatively low. This is because Forrester modeled the before state of the composite organization using data from Forrester's Developer Survey, 2023. This survey found that, on average, developers spend 13.5% of their time testing, 11.8% of their time debugging (only issues not related to security), and 7.5% of their time on nonproductive tasks. Because Forrester assumes that developers at the composite organization spend their time in the same ways before adopting GitLab, the benefit calculations are likely to be broadly applicable.

The most likely source of variability is differences in organizational characteristics — i.e., number of developers using GitLab and their average salaries. Variations in these characteristics will have predictable impacts on the results.

**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 $39.9 million.

> “The operational efficiency and productivity of [a major project] has [improved]. This is thanks to the collaboration tools. … There are thousands of people collaborating on this project, so having this platform certainly improves how they operate and how productive they are.”
>
> SOFTWARE ARCHITECT, ENERGY/RESEARCH

| Software Development Team Productivity | | | | | |
|---|---|---|---|---|---|
| **Ref.** | **Metric** | **Source** | **Year 1** | **Year 2** | **Year 3** |
| A1 | Software development team members using GitLab | R5 | 200 | 1,000 | 2,000 |
| A2 | Average time per user spent testing before GitLab | Forrester research | 13.5% | 13.5% | 13.5% |
| A3 | Reduction in testing time per user with GitLab | Interviews | 92% | 92% | 92% |
| A4 | Average testing time per user avoided with GitLab (hours) | A2*A3*2,080 | 258 | 258 | 258 |
| **A5** | **Subtotal: Testing time avoided after GitLab (hours)** | **A1*A4** | **51,600** | **258,000** | **516,000** |
| A6 | Average time per user spent debugging before GitLab (excluding time spent debugging security issues) | Forrester research | 11.8% | 11.8% | 11.8% |
| A7 | Average reduction in debugging time per user with GitLab | Interviews | 80% | 80% | 80% |
| A8 | Average debugging time per user avoided with GitLab (hours) | A6*A7*2,080 | 196 | 196 | 196 |
| **A9** | **Subtotal: Debugging time avoided after GitLab (hours)** | **A1*A8** | **39,200** | **196,000** | **392,000** |
| A10 | Average time per user spent on miscellaneous activities before GitLab | Forrester research | 7.5% | 7.5% | 7.5% |
| A11 | Reduction in miscellaneous activity time per user with GitLab | Interviews | 50% | 50% | 50% |
| A12 | Average miscellaneous activity time per user avoided with GitLab (hours) | A10*A11*2,080 | 78 | 78 | 78 |
| **A13** | **Subtotal: Miscellaneous activity time avoided after GitLab (hours)** | **A1*A12** | **15,600** | **78,000** | **156,000** |
| A14 | Fully burdened annual salary for a software development team member | Composite | $138,000 | $138,000 | $138,000 |
| A15 | Productivity recapture | TEI standard | 50% | 50% | 50% |
| At | Software development team productivity | (A5+A9+A13)*(A14/2,080)*A15 | $3,529,615 | $17,648,077 | $35,296,154 |
| | Risk adjustment | ↓10% | | | |
| Atr | Software development team productivity (risk-adjusted) | | $3,176,654 | $15,883,269 | $31,766,539 |
| | **Three-year total: $50,826,461** | | | **Three-year present value: $39,881,206** | |

## IMPROVED PRODUCTIVITY OF NEW HIRES TO THE SOFTWARE DEVELOPMENT TEAM

**Evidence and data.** After implementing GitLab Ultimate, interviewees explained that new developers were able to access code repositories, CI/CD pipelines, and collaborative tools more quickly, accelerating their onboarding. This streamlined environment shortened the ramp-up period to full productivity and enabled new hires to contribute meaningful work sooner. By automating routine tasks and providing clear, consistent processes, GitLab minimized onboarding friction so that new team members could quickly integrate into existing workflows and maintain high development velocity.

**75%**

Faster developer onboarding

- The program manager in finance explained: “Documentation plays an important role in getting people up to speed, and our web-based training and self-learning tools are very helpful for new users. When someone new joins our team, we encourage them to watch the training videos to familiarize themselves with our setup. These resources make it easy for new people to join the platform.”
- The CTO and SVP in defense said: “It’s community-driven support as well. It’s defining best practices and how to best use the tools. … If you want to come in and use a SaaS [software-as-a-service] tool, then the pointers for how to best integrate it into your DevSecOps pipeline [are in GitLab]. Just having 2,000 developers working together to push that state of the art forward helps a lot and makes it more efficient for [all] teams.”
- The software architect in energy/research noted: “Since GitLab is offered centrally and more and more teams are adopting GitLab into their workflows, it has become a very useful platform for the whole [organization]. … If a team is

well-organized — if they have good documentation and clear procedures — then the transition [for new team members] is easier. … The knowledge transfer to new colleagues [is easier].”

> “[GitLab creates] an ‘easy button’ for developers.”
>
> CTO AND SVP, DEFENSE

**Modeling and assumptions.** Based on the interviews, Forrester assumes the following for the composite organization:

- The software development team has an employee turnover rate of 15%. This is a reasonable estimate of employee turnover in the software development industry.[8]
- Without GitLab, it takes new developer hires 1.5 months on average to ramp up to full productivity, which is also normal in the industry.[9] New hires need time learn the tools, processes, and codebase of the composite organization.
- During the onboarding period, new hires are only 50% as productive as they are after they ramp up.
- With GitLab, new hires ramp up to full productivity 75% faster. By consolidating its toolchain with GitLab, the composite organization reduces the number of tools and simplifies the workflows that new hires must learn. In addition, integrated documentation and management features improve knowledge transfer.

**Risks.** This benefit is most likely to vary from organization to organization due to differences in employee turnover rates. Employee turnover rates can vary substantially across industries, roles, etc. Moreover, organizations currently face a unique macroeconomic environment and contemporary trends in employment may or may not reflect what organizations can expect over the long term.

For organizations with higher attrition or hiring rates, this benefit may be higher. To better isolate and show the impact of GitLab Ultimate, Forrester’s model holds total

headcount at the composite organization constant. While this is a best practice for Total Economic Impact models, most organizations might expect some growth — and thus potentially greater benefits — over three years.

Ultimately, though, this benefit accounts for only 3% of total benefits to the composite organization. While faster developer onboarding with GitLab Ultimate is important, the improvements in developer productivity (Benefit A) and time to market (Benefit C) are far more significant to the composite organization.

**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 $2.3 million.

| Improved Productivity Of New Hires To The Software Development Team | | | | | |
|---|---|---|---|---|---|
| **Ref.** | **Metric** | **Source** | **Year 1** | **Year 2** | **Year 3** |
| B1 | Total software development team members | R2 | 2,000 | 2,000 | 2,000 |
| B2 | Employee turnover rate of the software development team | Composite | 15% | 15% | 15% |
| B3 | New hires to the software development team per year | B1*B2 | 300 | 300 | 300 |
| B4 | New hires joining teams using GitLab | B3*R4 | 30 | 150 | 300 |
| B5 | Average time for a new hire to the software development team to become fully productive before GitLab (months) | Composite | 1.5 | 1.5 | 1.5 |
| B6 | Average productivity of a new hire while ramping to full productivity | Composite | 50% | 50% | 50% |
| B7 | Cost to the organization of a new hire's reduced productivity before GitLab | A14/12*B5*B6 | $8,625 | $8,625 | $8,625 |
| B8 | Reduction in time for a new hire to the software development team to become fully productive after GitLab | Interviews | 75% | 75% | 75% |
| Bt | Improved productivity of new hires to the software development team | B4*B7*B8 | $194,063 | $970,313 | $1,940,625 |
| | Risk adjustment | ↓5% | | | |
| Btr | Improved productivity of new hires to the software development team (risk-adjusted) | | $184,360 | $921,797 | $1,843,594 |
| | **Three-year total: $2,949,751** | | | **Three-year present value: $2,314,535** | |

## ACCELERATED TIME TO MARKET

**Evidence and data.** By offering a comprehensive DevSecOps platform that streamlined development, testing, and deployment processes, GitLab Ultimate enabled interviewees' organizations to become increasingly agile, releasing new features and updates into the market sooner than they otherwise would have. With faster release cycles of high-quality software, the interviewees' organizations were able to quickly respond to market demands and enhance their competitive edge, ultimately delivering features that drove customer satisfaction and incremental revenue gains.

Interviewees reported that their teams were delivering code to customers significantly faster since adopting GitLab. Examples included:

- One team using GitLab increased the frequency that it released to customers from once every four to six weeks to biweekly — delivering changes 33% to 50% faster.
- Teams using GitLab spun up new projects faster. Before GitLab, teams at one organization usually first deployed to production after three to five weeks. Using GitLab, though, the interviewees' organization streamlined project startup tasks (e.g., infrastructure provisioning, etc.) as well as standardized code and processes. Initial deployments to production now regularly happen on the first day of the project.
- The interviewees in the defense industry said that it takes their teams only one week to deploy and roll out software upgrades to their clients. This is significantly faster than other organizations in their industry can typically manage.

Time to first production release before GitLab

**15 to 25 days**

Time to first production release after GitLab

**<1 day**

Interviewees described their experiences with GitLab as follows:

- The CTO and SVP in defense explained: “When we had a more complicated tool stack, starting a new program and spinning up the stack to support that new program might take anywhere from three to six weeks. With GitLab, we’ve reduced the effort it takes to start new projects. We now tell our customers that we start on day one instead of week six.”

  The CTO and SVP continued: “Our team is able to deploy upgrades for our customers within a week of receiving them — something unheard of in our industry. If there’s an issue, we can roll back changes immediately, so we never experience an outage. We work 24/7 — our developers work all around the world and often on the weekends. ... Our developers [now] get new features up really quickly, rather than waiting five months. The power of GitLab software allows us to operate at this level.”

  They added: “We’re now able to get a clearer picture of software velocity because our organization is moving toward standardized software policies and metrics. … Before GitLab, we relied on a customer-provided environment, which was a mix of tools. Delivery was inconsistent — sometimes once a month, sometimes every six weeks. Since moving to GitLab and increasing automation, we’re now delivering updates and new features to customers every two weeks. Our customers know that they’re getting something new every two weeks rather than waiting indefinitely for the next release.”

> “[GitLab] is like our superpower. … Developers working on programs using Gitlab are responding to customer requirements faster and more nimbly.”
>
> **CTO AND SVP, DEFENSE**

**Modeling and assumptions.** Based on the interviews, Forrester assumes the following for the composite organization:

- Fifty percent of the composite organization's annual revenue is attributable to its proprietary software.
- Twenty percent of the annual revenues from software are due to new software development (e.g., new features, fixes, or update that either retain existing customers or attract new ones). The other 80% of software revenues come from revenue streams that do not require new development to maintain (e.g., recurring subscriptions to or sales of stable software).
- As a standard practice, the software development team releases every two weeks, and there are 52 work weeks in the year.
- Based on these assumptions, each developer's work during a release supports $9,615 in revenue.
- Thanks to efficiencies enabled by GitLab, the developers using GitLab deliver 50% of their annual releases 50% faster.
- In practice, the software development team still releases biweekly. However, aggregate output from the developers using GitLab is higher. These developers are delivering revenue-driving code changes — not all of their work, but some of it — sooner than they otherwise would have had they not been using GitLab. The amount of incremental work each developer is equivalent to 13 extra releases. In other words, in 26 releases, the developers using GitLab accomplish the same amount of work that previously took them 39 releases without GitLab.
- This enables the composite organization to realize some revenue from software developer sooner. Since the value of the work per developer per release is $9,615, the 13 annual incremental releases by each developer using GitLab drives about $25 million in incremental revenue in Year 1, about $125 million in incremental revenue in Year 2, and about $250 million in incremental revenue in Year 3. This is revenue shifted from later periods to the current years because the software development team is able to deliver faster.
- The composite organization has an operating margin of 15%. This is a normal operating margin for organizations in industries like those of the interviewees'

organizations. However, it is higher than the private sector market average, which is about 12%.[10]

## 3% higher revenue

Incremental revenue over three years from faster software delivery

**Risks.** This risk (or likelihood) of this benefit varying from organization to organization is relatively high. Although the framework for calculating this benefit is broadly applicable, organizations may realize substantially different results due to differences in:

- Percentage of total annual revenues driven by new software development.
- Operating margin.

Forrester encourages readers evaluating the potential benefits of faster software delivery and faster times to market to consider the corresponding metrics for their own organizations.

**Results.** To account for these risks, Forrester adjusted this benefit downward by 20%, yielding a three-year, risk-adjusted total PV (discounted at 10%) of $37.7 million.

| Accelerated Time To Market | | | | | |
|---|---|---|---|---|---|
| **Ref.** | **Metric** | **Source** | **Year 1** | **Year 2** | **Year 3** |
| C1 | Revenue (before GitLab) | Composite | $5,000,000,000 | $5,000,000,000 | $5,000,000,000 |
| C2 | Percentage of revenue attributed to software development | Composite | 50% | 50% | 50% |
| C3 | Percentage of software development revenue attributed to new software development (rather than recurring revenue) | Composite | 20% | 20% | 20% |
| C4 | Revenue from software development (before GitLab) | C1*C2*C3 | $500,000,000 | $500,000,000 | $500,000,000 |
| C5 | Total software development team members | R2 | 2,000 | 2,000 | 2,000 |
| C6 | Total available time per software development team member (weeks) | Composite | 52 | 52 | 52 |
| C7 | Release frequency before GitLab (weeks) | Composite | 2 | 2 | 2 |
| C8 | Software releases per developer before GitLab | C6/C7 | 26 | 26 | 26 |
| C9 | Revenue per release per developer | C4/C5/C8 | $9,615 | $9,615 | $9,615 |
| C10 | Percentage of releases accelerated with GitLab | Interviews | 50% | 50% | 50% |
| C11 | Reduction in delivery time for releases accelerated with GitLab | Interviews | 50% | 50% | 50% |
| C12 | Software releases per developer with GitLab | C6*(100%-C10)/ C7+C6*C10/(C7* (100%-C11)) | 39 | 39 | 39 |
| C13 | Incremental releases per software development team member after GitLab | C12-C8 | 13 | 13 | 13 |
| C14 | Software development team members using GitLab | R5 | 200 | 1,000 | 2,000 |
| C15 | Incremental revenue from software development team members using GitLab | C9*C13*C14 | $24,999,000 | $124,995,000 | $249,990,000 |
| C16 | Operating margin | Composite | 15% | 15% | 15% |
| Ct | Accelerated time to market | C15*C16 | $3,749,850 | $18,749,250 | $37,498,500 |
| | Risk adjustment | ↓20% | | | |
| Ctr | Accelerated time to market (risk-adjusted) | | $2,999,880 | $14,999,400 | $29,998,800 |
| | **Three-year total: $47,998,080** | | | **Three-year present value: $37,661,904** | |

## SECURITY EFFICIENCIES

**Evidence and data.** Interviewees highlighted that investing in GitLab Ultimate reduced the time and effort required for both security teams and developers to manage and mitigate security risks throughout the software development lifecycle.

- **Security team time savings.** GitLab Ultimate integrated robust security scanning, checking, and testing directly into the SDLC, automating these processes to ensure that vulnerabilities are identified and addressed early. Interviewees explained that this automation reduced the manual workload for their security teams, allowing them to focus on more strategic initiatives. In addition, GitLab Ultimate simplified the work of complying with regular security audits by providing comprehensive and easily accessible reports. Teams were able to meet external regulatory requirements with less effort.
- **Software development team time savings.** Interviewees explained that GitLab Ultimate enhanced developer efficiency by integrating comprehensive security and compliance tools throughout the SDLC. With features such as Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Dependency Scanning, developers could identify and resolve vulnerabilities early, reducing the risk of issues reaching production. They noted that automating security scans within CI/CD pipelines minimized manual intervention and the need for emergency patches. By proactively resolving issues in lower environments and utilizing Security Dashboards for enhanced visibility and collaboration, GitLab Ultimate helped the interviewees maintain stringent security standards with less effort, thereby significantly saving developer time and optimizing workflows.

Interviewees noted substantial productivity improvements around security and compliance-related activities with GitLab Ultimate:

| Security Efficiencies Reported By GitLab Customers | |
|---|---|
| **Activity** | **Example Successes** |
| Compliance | • **Faster auditing.** The external auditing process was shortened from several weeks to less than one week with GitLab automating 18 out of 25 quality criteria for internal assessments.<br>• **Faster reporting.** DevOps report creation, which previously required 6 hours per month, was automated with real-time dashboards in GitLab, reducing the time required to less than 1 hour.<br>• **Developer time savings.** Time spent creating a software bill of materials (SBOM) was eliminated, saving approximately 300 hours annually for the DevOps team. |
| Operations | • **Productivity gains.** Shifting left, automating processes, and improving workflows increased team productivity by 15%.<br>• **Productivity gains.** Security team efficiency for addressing vulnerabilities improved by 3%.<br>• **Productivity gains.** Shifting left saved approximately 22 FTEs.<br>• **Disaster prep efficiency.** Preparation time for disaster recovery fell from eight weeks with eight FTEs to two to three weeks with three FTEs.<br>• **Faster upgrades.** Time to upgrade pipelines fell from one week to between just 6 to 20 hours.<br>• **Faster scanning.** The entire security process — from moving files to scanning them to processing the results — was 13 times faster. |
| Incident response | • **Fewer issues.** New security findings in production environments decreased by 20% to 25%.<br>• **Faster responses.** Integrating security testing into pipelines reduced average response times from up to 30 days to just 1 hour. |

Mean response time before GitLab:

**30 days**

Mean response time after GitLab:

**1 hour**

At one organization, integrating security testing throughout development pipelines reduced average response times by 99%.

Interviewees also described their experiences with GitLab as follows:

- The program manager in finance explained: “Security is a top priority for us and our customers. Without our reputation [for security], we are nothing. A security breach could be disastrous — it could be the end of our company. GitLab has simplified the security process, bringing vulnerabilities closer to the team and integrating checks into the pipeline. By shifting left, automating processes, and improving workflows, we’ve seen significant improvements. [For example], about four weeks ago, our security team decided to perform manual code reviews on 5% of our low-risk programs. With GitLab, implementing that was very easy to do with just a few lines of code. If each team had to handle it in their own pipeline, it would have been horrible.”

> “Integrating security and quality scanners into the pipeline was a game changer for us. With more automation and less manual work, we’re seeing fewer failures, fewer problems, and faster progress. … [GitLab] has been critical for us.”
>
> **PROGRAM MANAGER, FINANCE**

- The software architect in energy/research shared: “We’ve always been focused on making our applications secure, but before GitLab, security processes were not well integrated in our teams’ processes. … Since moving to GitLab, we’ve made huge strides toward promoting security scanning and stronger security and compliance frameworks. All of these efforts help make our application more robust. … Our teams are more immersed in the security process. Now, from a single view in your application, you can see everything — commit changes, vulnerabilities, and more.”

  They added: “Our security team is [also] constantly creating documentation around security processes. … We have high turnover due to the nature of our

contracts, so it's important to ensure seamless knowledge transfer and maintain a culture of continuous improvement of security. … Before, if developers were not familiar with the security stack, they wouldn't spend time on it. Now, with all the tools and dashboard at their disposal, they dedicate more time to security. [And] because of how intuitive and accessible everything is [with GitLab], they're spending valuable time fixing issues more effectively."

- The CTO and SVP in the defense industry said: "Security has always been a priority for us, but with GitLab, we've shifted as far left as possible. … [Auditing and compliance] wasn't on our mind [when we invested in GitLab], but it's very much on our mind now as the government is moving toward regulating and securing the software supply chain. [Before GitLab], we were already doing DevSecOps; we were already doing integrated SAS and DAST and security check-ins. But there are additional requirements that come when we talk about securing the software supply chain. And GitLab was [already] thinking [about that] — from the SBOM to operating in an environment with tight controls."

  The product owner in defense added: "With GitLab, the security's on by default. … Previously, with a lot of programs, [the developers] were in the habit of just doing scans at the end. [They'd] build all the code and then at the end scan it before giving it to the customer. [But then they'd catch] something they'd need to fix. … Whereas in GitLab, [as a developer], you see [security] scan results every time you do a code commit — whether you want to or not. That leads developers to going, 'I should just fix this now.'"

- The senior consultant in finance also noted: "From an auditing and compliance perspective, we've started to use GitLab to make our processes smoother. We've established clear points in our pipelines where product owners and managers click approval processes to make sure we're capturing that data. This isn't just part of our change advisory board (CAB) process; it's also integrated into GitLab. Our end goal is for GitLab to serve as the primary source to demonstrate to auditors that we've met the evidence requirements."

> "Switching to GitLab helped us uncover a major blind spot in our previous scanning process that no one on our team was able to catch. It revealed a significant number of potential weaknesses now tracked as backlog items. We can now address new security findings in lower development environments, which helps us reduce overall costs. Overall, new security findings have dropped by 20% to 25% with GitLab."
>
> SENIOR CONSULTANT, FINANCE

**Modeling and assumptions.** Based on the interviews, Forrester assumes that with GitLab, security team members at the composite organization save time when 1) investigating and responding to incidents and 2) performing regular hardening tasks (e.g., preparing to recover from disasters, etc.). For each activity, Forrester makes the following specific assumptions:

- **For investigation and response time avoided with GitLab:**
  - Before GitLab, security team members spend about 6.7% of their annual work time investigating and responding to incidents.[11]
  - With GitLab, the amount of time security team members spend investigating and responding to incidents falls by 72%. Integrating automated security scanning throughout the software development lifecycle reduces the total number of incidents security team members need to respond to. As a unified platform, GitLab also makes it easier to trace, log, analyze, and manage incidents.
- **For disaster recovery prep time avoided with GitLab:**
  - Twenty percent of security team members are involved in disaster recovery prep.

- Before GitLab, these team members spend 15% of their annual work time on these activities.
- With GitLab, the amount of time security team members spend on disaster recovery prep falls by 88%.

Based on the interviews, Forrester also assumes that with GitLab, software development team members at the composite organization save time when performing security-related development activities including supporting auditing and compliance efforts and debugging security issues. For each activity, Forrester makes the following specific assumptions:

- **For auditing/compliance support time avoided with GitLab:**
  - Before GitLab, software development team members regularly spend about 3.85% of their annual work time supporting auditing and compliance efforts at the composite organization (e.g., fielding requests, documenting, etc.).
  - With GitLab, the amount of time software development team members spend supporting auditing and compliance work falls by 90% (i.e., due to improved visibility, etc.).
- **For security issue debugging time avoided with GitLab:**
  - Before GitLab, software development team members spend about 4.5% of the total time they spend debugging software focused specifically on security issues.
  - With GitLab, the amount of time software development team members spend debugging security issues falls by 20% (i.e., due to shifting left, etc.).

Finally, Forrester assumes the average fully burdened annual salary for a security team member is $116,000.[12] Like software development team members, security team members work 40 hours per week and capture and productively use 50% of the time they save with GitLab.

| Time Savings Per GitLab User | | | |
|---|---|---|---|
| **Team** | **Activity** | **Time Per Activity Annually Before Gitlab** | **Time Per Activity Annually After GitLab** |
| Security | Incident investigating and response | 139.4 hours | 39.0 hours |
| Security | Disaster recovery prep | 62.4 hours | 7.5 hours |
| **Total** | | **201.8 hours** | **46.5 hours** |
| Development | Auditing and compliance support | 80.1 hours | 8.0 hours |
| Development | Security issue debugging | 11.6 hours | 9.3 hours |
| **Total** | | **91.7 hours** | **17.3 hours** |

**Risks.** The risk (or likelihood) of this benefit varying from organization to organization is relatively low. This is because Forrester modeled the before state of the composite organization using data from Forrester's Security Survey, 2023, which found that, on average, security team members spend 6.7% of their time investigating and responding to cybersecurity incidents. Because Forrester assumes that team members at the composite organization spend their time similarly before adopting GitLab, the benefit calculations are likely to be broadly applicable.

However, this benefit captures only efficiencies the composite organization realizes while maintaining its current security posture after adopting GitLab. This benefit does not capture potential improvements in security posture (e.g., fewer incidents/issues, lower risk profile, etc.) The interviewees all said that their organizations' software was already extraordinarily secure and error-free before they started using GitLab. This was due to business requirements and partly a function of the industries the interviewees' organizations operated in. According to the interviewees, they could not feasibly raise software security or quality any higher. Without GitLab, though, achieving such high standards was extremely time-consuming, and for the interviewees, GitLab improved security by enabling them to maintain their exceptionally high standards more efficiently. Thus, organizations with different (i.e., more standard) business requirements for security and quality might see even greater benefits from GitLab due to additional improvements in security posture (e.g., reduced risk of breaches, etc.).

**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 $5.9 million.

| Security Efficiencies | | | | | |
|---|---|---|---|---|---|
| **Ref.** | **Metric** | **Source** | **Year 1** | **Year 2** | **Year 3** |
| D1 | Security team members using GitLab | R6 | 7 | 33 | 65 |
| D2 | Average time per user spent investigating and responding to security incidents before GitLab | Forrester research | 6.7% | 6.7% | 6.7% |
| D3 | Reduction in incident investigation and response time per user with GitLab | Interviews | 72% | 72% | 72% |
| D4 | Average incident investigation and response time per user avoided with GitLab (hours) | D2*D3*2,080 | 100 | 100 | 100 |
| **D5** | **Subtotal: Incident investigation and response time avoided after GitLab (hours)** | **D1*D4** | **700** | **3,300** | **6,500** |
| D6 | Percentage of security team members involved in disaster recovery prep | Interviews | 20% | 20% | 20% |
| D7 | Average time per user spent on disaster recovery prep before GitLab | Interviews | 15% | 15% | 15% |
| D8 | Reduction in disaster recovery prep time per user with GitLab | Interviews | 88% | 88% | 88% |
| **D9** | **Subtotal: Disaster recovery prep time avoided after GitLab (hours)** | **(D1*D6)*(2,080*D7*D8)** | **384** | **1,812** | **3,569** |
| D10 | Software development team members using GitLab | R5 | 200 | 1,000 | 2,000 |
| D11 | Average time per user spent supporting auditing/compliance before GitLab | Interviews | 3.85% | 3.85% | 3.85% |
| D12 | Reduction in time per user spent supporting auditing/compliance with GitLab | Interviews | 90% | 90% | 90% |
| D13 | Average time per user to support auditing/compliance avoided with GitLab (hours) | D11*D12*2,080 | 72 | 72 | 72 |
| **D14** | **Subtotal: Auditing/compliance support time avoided after GitLab (hours)** | **D10*D13** | **14,400** | **72,000** | **144,000** |
| D15 | Average time per user spent debugging before GitLab | Forrester research | 12.4% | 12.4% | 12.4% |
| D16 | Percentage of debugging time spent on security issues before GitLab | Interviews | 4.5% | 4.5% | 4.5% |
| D17 | Reduction in time per user spent addressing security issues with GitLab | Interviews | 20% | 20% | 20% |
| D18 | Average time per user to address security issues avoided with GitLab (hours) | D15*D16*D17*2,080 | 2 | 2 | 2 |
| **D19** | **Subtotal: Security issue debugging time avoided after GitLab (hours)** | **D10*D18** | **400** | **2,000** | **4,000** |
| D20 | Fully burdened annual salary for a security team member | Composite | $116,000 | $116,000 | $116,000 |
| D21 | Fully burdened annual salary for a software development team member | A14 | $138,000 | $138,000 | $138,000 |
| D22 | Productivity recapture | TEI standard | 50% | 50% | 50% |
| Dt | Security efficiencies | [(D5+D9)*(D20/2,080)+ (D14+D19)*(D21/2,080)] *D22 | $521,188 | $2,597,354 | $5,190,386 |
| | Risk adjustment | ↓10% | | | |
| Dtr | Security efficiencies (risk-adjusted) | | $469,069 | $2,337,619 | $4,671,347 |
| | **Three-year total: $7,478,035** | | **Three-year present value: $5,867,995** | | |

## TOOLCHAIN CONSOLIDATION

**Evidence and data.** Before investing in GitLab Ultimate, the interviewees' organizations managed sprawling toolchains. Tools often overlapped, and interviewees described their legacy toolchains as both inefficient and opaque. After switching GitLab, the interviewees noted their organizations realized direct cost savings as they retired now-redundant legacy tools. The interviewees' organizations benefited both by avoiding licensing costs for tools they no longer needed and by saving time on toolchain administration. For their organizations' IT departments, toolchains consolidated around GitLab were significantly easier to manage than the vast arrays of tools they had supported before.

Interviewees reported the following benefits from toolchain consolidation:

**Toolchain Consolidation Benefits Reported By Interviews**

| Cost Category | Example Successes |
|---|---|
| Software licensing costs | • GitLab replaced more than seven different tools, which the interviewees' organizations completely retired after the investment.<br>• Consolidating prior tools reduced total software licensing costs by 25% to 30%.<br>• Retiring redundant security tools saved $400,000.<br>• Sunsetted three out of the 13 core applications in the prior toolchain achieved a 23.1% reduction in the application set. |
| IT administration effort | • Toolchain management effort fell by 90%.<br>• Previously, six engineers were required to support the prior toolchain. Since consolidating the toolchain with GitLab, only three engineers were required.<br>• Saved one FTE by retiring a homegrown tool.<br>• Maintenance effort for a specific tool was reduced by 80%. |

> "We no longer need a patchwork of open-source software pieces cobbled together. We didn't even do a formal analysis of alternatives because GitLab was first to market with a platform that does everything we need. We could quickly see that consolidating onto GitLab would eliminate five to seven tools from any given program."
>
> CTO AND SVP, DEFENSE

Interviewees further described their experiences with GitLab as follows:

- The supervisor in finance explained: “After GitLab, the engineering headcount required to both support and maintain our tools decreased. This allowed us to focus more on supporting our end users rather than just maintaining infrastructure. We also no longer have to handle in-house upgrades, which used to take a lot of effort. We’re also avoiding issues around poor documentation and turnover within the team that developed our homegrown tools, which caused a major loss of knowledge and made support even more labor-intensive.”
- The CTO and SVP in defense said: “The big win for us was efficiency — both in administration and in overall operations. Now, everyone can work collaboratively, and we can easily automate our pipeline. I’m also able to move personnel around to complete different tasks more efficiently. Rather than needing to train on different tools across programs, now it’s just ‘learn GitLab,’ and they’re ready to begin working.”

**Modeling and assumptions.** Based on the interviews, Forrester assumes the following for the composite organization:

- Before GitLab, the composite organization’s IT budget is 2.4% of annual revenue and it spends 19% of its IT budget on licensing costs for third-party software to support business operations (versus on other IT costs such as hardware, IT employee salaries, etc.). Such spending levels are typical for organizations like the composite.[13]
- Tools for software development and security constitute about 50% of the composite organization’s annual spending on software.
- In addition, there are 25 IT administrators who, before GitLab, each spend around 80% of their time supporting and managing this tooling (i.e., managing updates, supporting users, etc.).
- As the composite organization rolls out GitLab across its software development and security teams, users migrate from their legacy point-solution tools to features built into the GitLab platform. This allows the composite organization to gradually reduce its spending on legacy tooling. Costs fall as the organization purchases fewer licenses and/or sunsets redundant tools. In total, consolidating on GitLab reduces toolchain software costs by 25%.

**25%**

Reduction in software toolchain costs

- For the composite organization, the potential savings from tool consolidation are even greater than 25% but in practice, some legacy spending lingers on even after teams consolidate on GitLab (e.g., long-term contracts lock in spending; busy IT admins require more time to fully retire tools; certain tools remain in occasional use occasionally because they fulfill a specialized need; etc.)
- As the composite organization consolidates on GitLab, the IT administrators responsible for the software toolchains save time as well. They have fewer tools to manage and toolchain administration in GitLab is both easier and more efficient than in legacy tools. On the whole, migrating to GitLab reduces corresponding toolchain administration effort by 75%.

**75%**

Reduction in IT effort to administer toolchain

- Finally, the average fully burdened annual salary for an IT administrator is $125,000.[14] IT administrators also work 40 hours per week and capture and productively use 50% of the time they save after GitLab.

**Risks.** The risk (or likelihood) of this benefit varying from organization to organization is relatively low. This is because Forrester modeled the before state of the composite organization using data from Forrester's 2024 IT And Digital Budget Benchmarks. The benefit calculations are likely to be broadly applicable.

**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 $4.3 million.

> “Removing our toolchain saved context switching. Project management, code suggestions, continuous integration, and more are now all combined into one platform.”
>
> SOFTWARE ARCHITECT, ENERGY/RESEARCH

| Toolchain Consolidation | | | | | |
|---|---|---|---|---|---|
| **Ref.** | **Metric** | **Source** | **Year 1** | **Year 2** | **Year 3** |
| E1 | Percentage of revenue allocated to IT budget (before GitLab) | Forrester research | 2.4% | 2.4% | 2.4% |
| E2 | Percentage of IT budget allocated to software (before GitLab) | Forrester research | 19% | 19% | 19% |
| E3 | Percentage software budget allocated to licensing costs of software development and security tools (before GitLab) | Composite | 50% | 50% | 50% |
| E4 | Licensing costs for software development and security toolchain before GitLab | C1*E1*E2*E3 | $11,400,000 | $11,400,000 | $11,400,000 |
| E5 | Percentage of software development and security team members using GitLab | R4 | 10% | 50% | 100% |
| E6 | Reduction in licensing costs for development and security toolchain by retiring tools with GitLab | Interviews | 25% | 25% | 25% |
| **E7** | **Subtotal: Toolchain licensing costs avoided after GitLab** | **E4*E5*E6** | **$285,000** | **$1,425,000** | **$2,850,000** |
| E8 | Toolchain administrators | Composite | 25 | 25 | 25 |
| E9 | Time spent on toolchain administration before GitLab (hours) | E8*2,080*80% | 41,600 | 41,600 | 41,600 |
| E10 | Reduction in toolchain administration time with GitLab | Interviews | 75% | 75% | 75% |
| **E11** | **Subtotal: Toolchain administration time avoided after GitLab (hours)** | **E9*E5*E10** | **3,120** | **15,600** | **31,200** |
| E12 | Fully burdened annual salary for an IT admin | Composite | $125,000 | $125,000 | $125,000 |
| E13 | Productivity recapture | TEI standard | 50% | 50% | 50% |
| Et | Toolchain consolidation | E7+E11*(E12/2,080) *E13 | $378,750 | $1,893,750 | $3,787,500 |
| | Risk adjustment | ↓10% | | | |
| Etr | Toolchain consolidation (risk-adjusted) | | $340,875 | $1,704,375 | $3,408,750 |
| | **Three-year total: $5,454,000** | | | **Three-year present value: $4,279,505** | |

## UNQUANTIFIED BENEFITS

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

- **Better developer experience.** Interviewees reported that GitLab Ultimate enhanced developer satisfaction at their organizations by integrating all essential development tools into a single platform, eliminating the frustration of juggling disparate systems. Its streamlined workflows and automated CI/CD pipelines reduced repetitive tasks and manual interventions, freeing developers to focus on more meaningful work. They also reported that the enhanced security and

compliance features mitigated the stress of managing vulnerabilities and adhering to regulations. Overall, GitLab Ultimate's cohesive, efficient, and user-friendly experience led to a more enjoyable and fulfilling development process.

The product owner in defense said: "Within GitLab, we've built a hub of common code and other resources to make sure our developers have what they need to get things done. The templates and code are ready to go so they can start right away. This helps create a sense of community."

> "There has been a rise in collaboration. The fact that you have now a tool in which you can discuss — [in which] you can do everything [you need to] to improve your software — is very useful."
>
> SOFTWARE ARCHITECT, ENERGY/RESEARCH

- **Superior support.** Interviewees noted that GitLab Ultimate's customer support and partnership helped them realize the full value of the platform. With features like 24/7 global support, dedicated customer success managers, and tailored onboarding programs, GitLab minimized any disruptions to software development and supported the interviewees as they built business-critical software.

  The interviewees also mentioned that their partnership with GitLab gave them access to proactive guidance on optimizing CI/CD pipelines, automating workflows, and implementing security best practices. Regular check-ins and access to exclusive resources kept the interviewees informed about new features and industry trends, enabling them to continuously maximize their GitLab usage and stay ahead in their industries.

  The program manager in finance said: "When we first implemented GitLab, we faced some internal challenges. GitLab immediately showed their commitment by

saying, ‘Your success is our success. We’re here for you.’ When we fell behind schedule, I escalated the issue and within two or three days, our GitLab sales rep took action, bringing in the right people to address our problems — everyone from developers to customer success, management, and sales representatives stepped up. The GitLab Advocacy Program also played a crucial role, with a dedicated contact for our region who checked in regularly, offered help, and provided valuable resources. This level of support and flexibility made a big difference.”

> “Among all the vendors we work with, GitLab stands out to us for its partnership. They meet with us regularly to spend time helping us with engineering, specific features, and more. When we mention a specific feature that we need, they spend time to listen and work with us to make it happen.”
>
> PRODUCT OWNER, DEFENSE

## FLEXIBILITY

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

- **GitLab Duo AI features.** Interviewees planned to increasingly use GitLab Ultimate’s AI capabilities — known as GitLab Duo — to integrate AI throughout their workflows. Interviewees believed that functions like intelligent code suggestions and optimization could further accelerate their development cycles and improve code quality. Additionally, they expected Duo’s real-time support and contextual guidance to enable faster problem-solving and reduce time spent debugging.

The software architect in energy/research explained: “Our developers are eager to begin using GitLab’s AI-powered tools because of all the benefits they can bring to our organization. For instance, if a developer needs to code something and can’t remember how, they’d typically have to spend time [researching]. GitLab has built-in tools that work like an advisor for you. Features like code suggestions help provide ideas. It can also help optimize code and answer questions — these things will help achieve additional time savings.”

The software architect described their initial experience testing GitLab Duo: “The functionality is looks very promising. For instance, generating a CI image typically takes a half-hour or more — now it’s done in just seconds. Simply copy, paste, and it works.”

The consultant in defense also shared: “We’re definitely leaning toward integrating generative AI into our software development practices and want GitLab as partner in that effort. Our hope is that, by using GitLab Duo, we can become more efficient as a company and even increase developer satisfaction.”

They continued: “We’re in the process of looking at our existing AI-powered tools and working with GitLab to eventually retire those solutions and fill in those gaps of functionality with GitLab Duo. We currently have seven to eight different use cases that we want to leverage GitLab’s AI features for. As we continue to grow our need for AI tools, GitLab Duo will continue to expand to support additional features.”

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

## 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** |
| Ftr | GitLab Ultimate licensing costs | $0 | $270,508 | $1,349,924 | $2,698,542 | $4,318,974 | $3,389,011 |
| Gtr | Implementation | $588,500 | $117,832 | $58,916 | $0 | $765,248 | $744,311 |
| Htr | Ongoing management | $0 | $85,800 | $429,000 | $858,000 | $1,372,800 | $1,077,174 |
| Itr | On-premises infrastructure | $0 | $434,700 | $2,169,300 | $4,336,500 | $6,940,500 | $5,446,068 |
| Jtr | New user training | $0 | $601,128 | $2,398,704 | $2,996,928 | $5,996,760 | $4,780,516 |
| | Total costs (risk-adjusted) | $588,500 | $1,509,968 | $6,405,844 | $10,889,970 | $19,394,282 | $15,437,080 |

### GITLAB ULTIMATE LICENSING COSTS

**Evidence and data.** Interviewees reported that GitLab's scalable pricing model, which adjusts according to user count and feature needs, enabled them to manage costs efficiently while supporting their growth. This flexibility ensured that as their requirements evolved, their investment in GitLab remained aligned with their budget and expansion goals.

The CTO and SVP in defense said: "Based on our research, GitLab aligned with our vision. With [other competitors], we would have had to negotiate complex licenses. We wanted to avoid that complexity, and GitLab provided a straightforward solution."

**Modeling and assumptions.** Forrester assumes the composite organization's licensing cost per user for GitLab Ultimate is $99 per month ($1,188 per year).

**Risks.** The risk (or likelihood) of this cost varying from organization to organization is relatively low. Pricing for GitLab Ultimate might vary depending on a variety of factors (e.g., feature and usage requirements). Readers should contact GitLab for additional details.

**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 $3.4 million.

| GitLab Ultimate Licensing Costs | | | | | | |
|---|---|---|---|---|---|---|
| **Ref.** | **Metric** | **Source** | **Initial** | **Year 1** | **Year 2** | **Year 3** |
| F1 | Total GitLab users | R5+R6 | | 207 | 1,033 | 2,065 |
| F2 | Cost per user for GitLab Ultimate license | Composite | | $1,188 | $1,188 | $1,188 |
| Ft | GitLab Ultimate licensing costs | F1*F2 | | $245,916 | $1,227,204 | $2,453,220 |
| | Risk adjustment | ↑10% | | | | |
| Ftr | GitLab Ultimate licensing costs (risk-adjusted) | | $0 | $270,508 | $1,349,924 | $2,698,542 |
| | **Three-year total: $4,318,974** | | | **Three-year present value: $3,389,011** | | |

## IMPLEMENTATION

**Evidence and data.** Interviewees described land-and-expand approaches to deploying GitLab, i.e., they started with core use cases and then scaled their usage of the platform — in terms of both functionality and user count — over time. They leveraged both internal resources and professional services for their initial implementation periods, which took months to complete. However, interviewees noted that in subsequent years, expanding their GitLab usage required significantly less effort.

The program manager in finance said: “It took [several] months to customize the platform before we started using it with the first team. We then expanded to three and eventually six teams. We had two to four GitLab professional services team members help with implementation and customization. We also worked with a GitLab customer success team member who helped us ask the right questions and focus on key areas. It was a very helpful part of our implementation journey.”

The program manager continued: “We approached implementation in our usual way. We worked with a mixed team of internal developers and external members to speed up the process and provide us with the expertise we needed. With this method, we were able to scale faster.”

**Modeling and assumptions.** Based on the interviews, Forrester assumes the following for the composite organization:

- Initial deployment of GitLab Ultimate takes 6,240 hours in total. This work is divided amongst multiple IT administrators at the composite organization (e.g., six employees working for six months).
- To help with the initial deployment, the composite organization also spends $160,000 on professional services from either GitLab or a third party.
- The composite organization also incurs implementation costs — both IT admin effort and professional services — in subsequent years in order to expand its GitLab usage. Such implementation costs are leading, i.e., to support GitLab usage in Year 2, the composite invests in Year 1 ahead of time. The composite has no implementation costs in Year 3 because by then, GitLab is fully adopted throughout the organization.

- Implementation costs fall year to year: as more of the organization migrates to GitLab, it becomes increasingly easier to add new users, use cases, etc. The rate at which the implementation costs fall is inversely proportional to the rate at which new users are added.

**Risks.** The risk (or likelihood) of this cost varying from organization to organization is moderate. The interviewees reported similar implementation costs when adjusted (or normalized) for various characters (e.g., number of users, etc.). However, the more that an organization differs from the composite (e.g., in terms of size, industry, business model, etc.), the more likely it is that the organization may experience a different scale of implementation costs — either disproportionately higher or lower. Implementation costs can also be affected by an organization's existing IT environment (e.g., from overall IT complexity and business constraints to specific tools in the legacy toolchain). Lastly, some organizations may opt to substitute internal IT implementation effort for additional professional services (or vice versa). The implementation costs modeled — both internal IT effort and professional services — are representative of the experiences of the interviewees' organizations only. The costs are not benchmarks, and readers should consult both their internal teams and GitLab when estimating their own professional services costs.

**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 $744,000.

| Implementation | | | | | | |
|---|---|---|---|---|---|---|
| **Ref.** | **Metric** | **Source** | **Initial** | **Year 1** | **Year 2** | **Year 3** |
| G1 | Total implementation time from toolchain administrators (hours) | Interviews | 6,240 | 1,250 | 625 | 0 |
| G2 | Professional services | Interviews | $160,000 | $32,000 | $16,000 | $0 |
| Gt | Implementation | G1*(E12/2,080)+G2 | $535,000 | $107,120 | $53,560 | $0 |
| | Risk adjustment | ↑10% | | | | |
| Gtr | Implementation (risk-adjusted) | | $588,500 | $117,832 | $58,916 | $0 |
| | **Three-year total: $765,248** | | | **Three-year present value: $744,311** | | |

## ONGOING MANAGEMENT

**Evidence and data.** Interviewees reported updating GitLab Ultimate more frequently than other platforms, but they described overall management effort as relatively minimal due to self-service capabilities that enabled users to troubleshoot and resolve issues without IT administrator support.

- The program manager in finance explained: “Compared to our old system, GitLab requires that we do more frequent updates. Since it runs 24/7, we have five resources that split their time between managing GitLab along with our mainframe pipeline and [another tool] in our environment.”
- The product owner in defense shared: “We update GitLab on a weekly basis — I report on that as part of my status update. Every week, we spend a few hours contributing to our documentation and shared code repository. It’s relatively low effort, but we want to make sure we update documentation if we’re deploying a new feature or marking changes.”

**Modeling and assumptions.** Based on the interviews, Forrester assumes the following for the composite organization:

- The composite organization still has a total of 25 IT administrators who manage its toolchain (in addition to performing other tasks).
- The number of full-time or equivalent (FTE) IT administrators supporting GitLab grows as the organization rolls out GitLab to more users (an FTE could be a single employee who spends all their time on one activity, or an FTE could consist of multiple employees whose combined time spent on an activity is the equivalent of one employee fully dedicated to it).
- The IT admin FTEs supporting GitLab regularly spend about 25% of their time on the platform.

**Risks.** The risk (or likelihood) of this cost varying from organization to organization is relatively low. The most likely source of variability is in the amount of time toolchain administrators spend managing GitLab, which could vary based on deployment characteristics (e.g., scale/scope or features deployed), user support needs (e.g., for documentation or ongoing support), business requirements (e.g., enhanced security needs), or other reasons.

**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 $1.1 million.

| Ongoing Management | | | | | | |
|---|---|---|---|---|---|---|
| **Ref.** | **Metric** | **Source** | **Initial** | **Year 1** | **Year 2** | **Year 3** |
| H1 | Toolchain administrators | E8 | | 25 | 25 | 25 |
| H2 | Percentage of software development and security team members using GitLab | R4 | | 10% | 50% | 100% |
| H3 | Percentage of toolchain administrator time allocated to ongoing management of GitLab for users | Interviews | | 25% | 25% | 25% |
| **H4** | **Subtotal: Toolchain administrator time spent on ongoing management of GitLab (hours)** | **H1*H2*H3*2,080** | | **1,300** | **6,500** | **13,000** |
| Ht | Ongoing management | H4*(E12/2,080) | $0 | $78,000 | $390,000 | $780,000 |
| | Risk adjustment | ↑10% | | | | |
| Htr | Ongoing management (risk-adjusted) | | $0 | $85,800 | $429,000 | $858,000 |
| | **Three-year total: $1,372,800** | | | **Three-year present value: $1,077,174** | | |

## ON-PREMISES INFRASTRUCTURE

**Evidence and data.** For business reasons, the interviewees' organizations opted to run GitLab Ultimate from on-premises infrastructure rather than via the public cloud. This created some additional ongoing costs associated with the platform. While the interviewees were comfortable with and had chosen GitLab in part because of its security, their organizations operated in business environments that necessitated on-premise deployments (e.g., for the organization in the defense industry, contracts often required on-premises tools for security reasons; for the organization in the energy and research space, maintaining a private cloud both enhanced security and reduced latency, etc.).

**Modeling and assumptions.** Based on the interviews, Forrester assumes the on-premises IT infrastructure that the composite organization uses to run GitLab Ultimate costs the organization about $2,000 per GitLab user per year. Total costs thereby increase as the organization expands its GitLab usage and deployment.

**Risks.** The risk (or likelihood) of this cost varying from organization to organization is moderate. The business reasons spurring the interviewees to run GitLab on-premises may not apply to many other organizations. While the interviewees' usage of GitLab Ultimate to build software for the most demanding business needs is a testament to the strength of the GitLab platform, other organizations may not face such stringent requirements. Forrester accounts for on-premises infrastructure costs in this study both to be conservative and to accurately report the interviewees' experiences. However, in contrast to the interviewees' organizations, many organizations are likely to use GitLab Ultimate via the public cloud and thus avoid on-premises infrastructure costs altogether. Organizations that do may realize lower costs and higher ROIs than the composite.

**Results.** To account for these risks, Forrester adjusted this cost upward by 5%, yielding a three-year, risk-adjusted total PV (discounted at 10%) of $5.4 million.

| On-Premises Infrastructure | | | | | | |
|---|---|---|---|---|---|---|
| **Ref.** | **Metric** | **Source** | **Initial** | **Year 1** | **Year 2** | **Year 3** |
| I1 | Total GitLab users | F1 | | 207 | 1,033 | 2,065 |
| I2 | Approximate on-premises infrastructure cost per GitLab user | Interviews | | $2,000 | $2,000 | $2,000 |
| It | On-premises infrastructure | I1*I2 | $0 | $414,000 | $2,066,000 | $4,130,000 |
| | Risk adjustment | ↑5% | | | | |
| Itr | On-premises infrastructure (risk-adjusted) | | $0 | $434,700 | $2,169,300 | $4,336,500 |
| | **Three-year total: $6,940,500** | | | **Three-year present value: $5,446,068** | | |

## NEW USER TRAINING

**Evidence and data.** Interviewees said that GitLab Ultimate's intuitive interface and extensive documentation flattened the learning curve for new users.

- The program manager in finance said: "Transitioning our teams to the new system took time because they wanted to feel confident. … Even though our new pipeline is better than our old system, people need time to adjust to a new way of doing things after years of working in a specific way."

  The program manager continued: "All users had a five-day training period, which included agile and DevSecOps training in addition to specialized security sessions. We also offered some deep-dive training for more advanced users. The training sessions were a good opportunity for people in different departments — including those in developer, operations, and business roles — to collaborate after being siloed for many years."

> "GitLab is an intuitive product. We're finding we don't get a lot of requests for trainings because people are able to immerse themselves in it and go. People are stunned at how easy it is to use."
>
> CTO AND SVP, DEFENSE

- The product owner in defense shared: “Trainings were provided to us as part of our partnership with GitLab. They hosted many different sessions that we encouraged people to join. They also came to our meetings to do presentations and answer questions, which helped with community building. The more people that use GitLab, the more people are available in the community to help each other.”

  The product owner also said: “We run four rounds of our introductory training. It’s a week-long course in which we cover what GitLab is, how to use it, its security features, and more. After that training, users are able to start using the tool.”

  The product owner concluded: “Becoming an expert GitLab is a journey. You learn from your team, colleagues, and resources such as advanced webinars. Everyone in our company has access to on-demand training provided by GitLab. Many people on our team use it as a resource to learn in their own time.”

**Modeling and assumptions.** Based on the interviews, Forrester assumes users at the composite organization who are new to GitLab complete 40 hours (five full workdays) of training. This training is flexible and diverse (e.g., webinars, videos, documentation, discussions with colleagues; introductory as well as advanced materials; scheduled/required sessions plus self-motivated learning; etc.) but totals up to 40 hours per user overall.

**Risks.** The risk (or likelihood) of this cost varying from organization to organization is relatively low. The most likely source of variability might be users’ prior experience with GitLab; users may need more or less training accordingly.

In practice, new user training takes a variety of forms: organizations may offer formal, required training sessions; users may research solutions to their problems on their own; users may learn from their colleagues; organizations may develop their own training materials and/or they may contract with GitLab or third-parties for training, etc. When this training occurs can vary as well: some training may be necessary to get started and then some learning may occur months or even years later. Forrester’s model for the composite organization accounts for all this variability. Based on the interviews, Forrester assumes that new GitLab users become fully proficient after spending a total of 40 hours learning the platform. The format and schedule of that training can vary

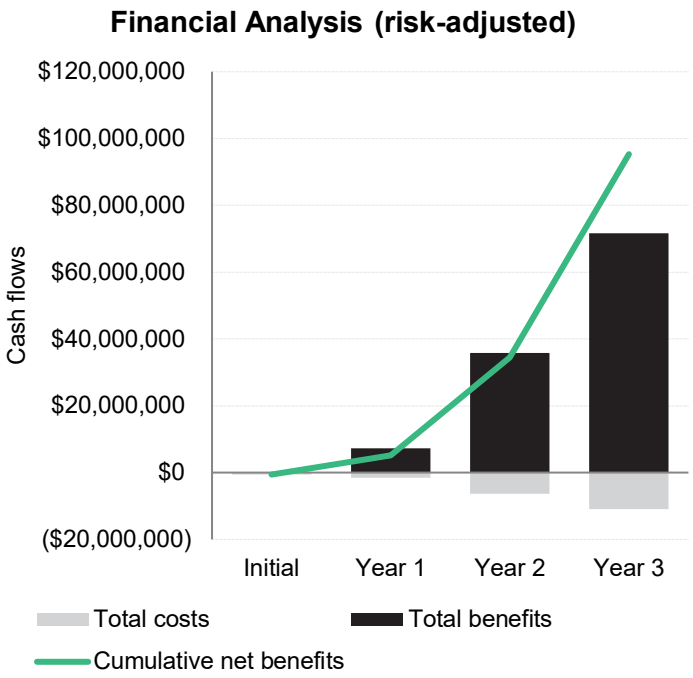
significantly, but the composite organization still incurs the cost of its employees' time until those employees become fully proficient with a new platform.

**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 $4.8 million.

| New User Training | | | | | | |
|---|---|---|---|---|---|---|
| **Ref.** | **Metric** | **Source** | **Initial** | **Year 1** | **Year 2** | **Year 3** |
| J1 | Total GitLab users | F1 | | 207 | 1,033 | 2,065 |
| J2 | New GitLab users | J1-J1$_{PY}$ | | 207 | 826 | 1,032 |
| J3 | Training time per new GitLab user (hours) | Interviews | | 40 | 40 | 40 |
| J4 | Fully burdened hourly rate for a new GitLab user | [(R5*A14+R6*D20)/ (R5+R6)]/2,080 | | $66 | $66 | $66 |
| Jt | New user training | J2*J3*J4 | $0 | $546,480 | $2,180,640 | $2,724,480 |
| | Risk adjustment | ↑10% | | | | |
| Jtr | New user training (risk-adjusted) | | $0 | $601,128 | $2,398,704 | $2,996,928 |
| | **Three-year total: $5,996,760** | | | **Three-year present value: $4,780,516** | | |

# 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 | ($588,500) | ($1,509,968) | ($6,405,844) | ($10,889,970) | ($19,394,282) | ($15,437,080) |
| Total benefits | $0 | $7,170,838 | $35,846,460 | $71,689,030 | $114,706,328 | $90,005,145 |
| Net benefits | ($588,500) | $5,660,870 | $29,440,616 | $60,799,060 | $95,312,046 | $74,568,065 |
| ROI | | | | | | 483% |
| 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."

### PRESENT VALUE (PV)

The present or current value of (discounted) cost and benefit estimates given at an interest rate (the discount rate). (This study uses a discount rate of 10% per year.) 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.

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.

## APPENDIX B: 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.

[2] Forrester finds that developers save a total of 609 hours per year with GitLab due to improvements throughout the SDLC (see Benefit A: Software Development Team Productivity and Benefit D: Security Efficiencies). Forrester then conservatively assumes that half of that time saved is used productively (i.e., on work generating business value). Some of the time developers save is consumed by less-productive activities (e.g., meetings, coffee breaks, chatting with coworkers, etc.).

[3] Forrester finds that security team members save a total of 155 hours per year with GitLab (see Benefit D: Security Efficiencies). Forrester then conservatively assumes that half of that time saved is used productively (i.e., on work generating business value). Some of the time saved is consumed by less-productive activities (e.g., meetings, coffee breaks, chatting with coworkers, etc.).

[4] Source: Forrester's Developer Survey, 2023.

[5] Ibid.

[6] Ibid.

[7] Source: TIER: US Software Developer Labor Market, 2024, Forrester Research, Inc., June 10, 2024.

[8] Sources: Evgenia Kuzmenko, The Costs of Quiet Quitting: Hidden Consequences of Employee Turnover in Software Development, Kitrum, May 2, 2023; Greg Lewis, Industries with the Highest (and Lowest) Turnover Rates, LinkedIn, August 11, 2022.

[9] Source: John Hall, The Cost Of Turnover Can Kill Your Business And Make Things Less Fun, Forbes, May 9, 2019.

[10] Source: Aswath Damodaran, Margins by Sector (US), NYU Stern, January 2024.

[11] Source: Forrester's Security Survey, 2023.

[12] Source: Average Salary for People with Jobs in Computer/Network Security, Payscale, August 2024.

[13] Source: 2024 IT And Digital Budget Benchmarks, North America, Forrester Research, Inc., May 21, 2024.

[14] Source: Average Information Technology (IT) Manager Salary, Payscale, August 2024.

FORRESTER®