



Eliminate the 5 bottlenecks killing your delivery speed

Learn how your team can ship smarter and faster





Table of Contents

/03/	The scaling paradox: Why more can mean less
/06/	The 5 common bottlenecks growing teams face • Manual development overhead • Merge coordination breakdown • Development workflow blindspots • The planning paralysis • Integrated Agile project management
/17/	Scale your development process: The cost of standing still

The scaling paradox: Why more can mean less

As your team grows, you add more developers, tools, and processes. In theory, this should make your team more productive. In reality, as teams grow from 5 to 50 to 500 developers, the complexity of coordination increases tenfold.

What once required a quick conversation between teammates evolves into formal meetings, documentation requirements, and multi-layer approval processes that consume more time than they save.

The same pattern emerges with AI tools: teams add AI assistants expecting productivity gains, but these tools work in isolation — understanding individual files but not your deployment patterns, security policies, or team standards. You're left with a vicious cycle where each solution to manage complexity introduces new complexity that requires even more management.

Whether you're a startup growing from 5 to 50 developers or an enterprise navigating the complexities of a global team, scaling while maintaining productivity is a challenge teams of all sizes face. Learn how to streamline your processes and eliminate the bottlenecks preventing your organization from its full potential.



Why do workflows become so complex?

There’s a well-known calculation among project managers that analyzes the number of communication paths in relation to team size. As teams grow, the number of communication paths grows exponentially.

In the calculation below, n is the number of people on the team. A team with 5 members has 10 communication paths. When the team grows to 25 people, its communication paths expand to 30 times that of the original 5-person team.

The calculation assumes that each team member has direct communication with every other team member. While this isn’t necessarily the case, the calculation offers a glimpse at the complexity organizations face as they scale. As team size grows, so do other pain points, such as a lack of visibility, knowledge sharing between teams, and tool sprawl.

$$\frac{n*(n-1)}{2} = X$$

$$\frac{5*(4)}{2} = 10$$

$$\frac{25*(24)}{2} = 300$$

$$\frac{200*(199)}{2} = 19,900$$

Team Size	Communication Paths	Pain Points
5-person startup	10 paths	The cost of technical debt Knowledge transfer from founders to new team members
25-person scale-up	300 paths	Communication overhead Infrastructure growing pains
200-person enterprise	19,900 paths	Knowledge silos between teams Toolchain sprawl



Identify the systems that enable you to scale

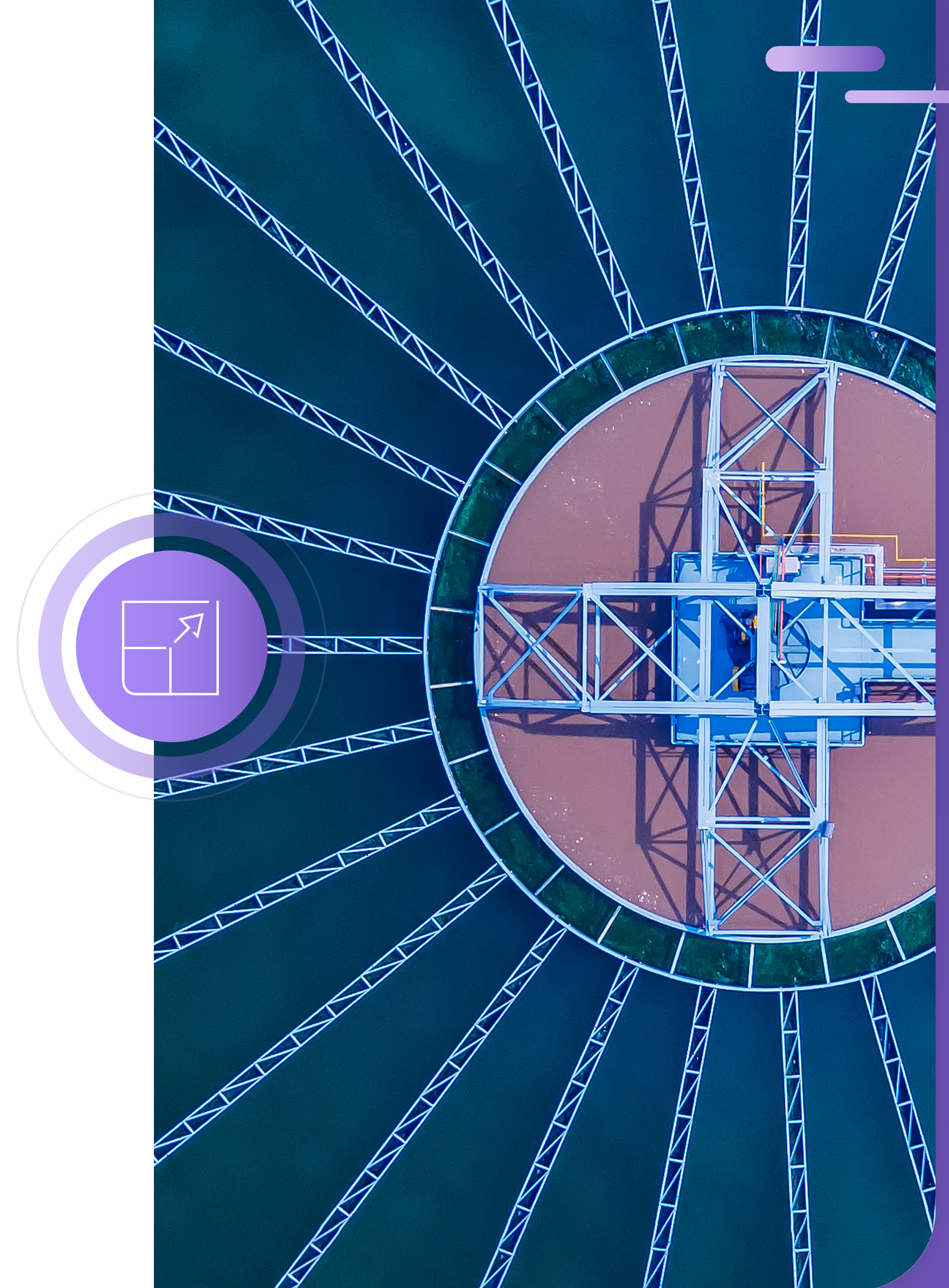
The most successful growing teams don't fight the scaling paradox — they architect around it.

They understand that scale isn't just about adding people; it's about designing systems, processes, and culture that maintain small-team effectiveness at large-team capacity.

This means making deliberate choices about when to add structure and when to preserve flexibility. It means investing in automation that eliminates coordination overhead rather than managing it. Most importantly, it means recognizing that the practices and free tools that got you here won't necessarily take you to the next level.

While free tools are a launchpad for teams entering the world of DevSecOps, they're best suited for small teams and individual developers. As your systems become more complex, your tools should evolve to meet your growing needs.

The following bottlenecks represent the most common places where the scaling paradox strikes — and how leveraging GitLab Premium can help you scale up without slowing down.



The 5 common bottlenecks growing teams face



/01/ Manual development overhead



/02/ Merge coordination breakdown



/03/ Development workflow blindspots



/04/ The planning paralysis



/05/ Troubleshooting without support





/01/

Manual development overhead

The problem: Routine tasks consume development capacity

Our most recent survey showed that 21% of developer time is spent writing new code, 15% is spent improving existing code, 15% on meetings and administrative tasks, and 13% is spent understanding what code does.

Teams often try to accelerate workflows by adding AI coding assistants, but these tools typically only help with one task: new code creation. The other activities — code reviews, understanding legacy systems, coordinating deployments — require the same manual effort because the AI tools work in isolation from your broader development workflow.

If teams were able to streamline some of those tasks, developers would be able to focus on more projects that push innovation rather than maintaining the status quo.



Teams aren't reaching their full potential

Development teams hit productivity ceilings not because they lack skill, but because routine tasks consume the time and mental energy that could drive innovation.



Knowledge gaps

As codebases grow and teams expand, new team members need more support to understand existing code and legacy systems.



Distributed team friction

When teams work across time zones or need to collaborate on unfamiliar code, knowledge transfer becomes a significant bottleneck. Straightforward projects become extensive debugging and explanation sessions.

Percentage of time developers spend on daily tasks¹



¹ GitLab, 2024 Global DevSecOps Report

The solution: AI that supports the full development lifecycle

To effectively leverage AI across workflows, teams need AI that understands every aspect of the software development lifecycle (SDLC). GitLab Duo is our comprehensive AI suite that includes **Chat**, **Code Suggestions**, and **Duo Agent Platform**, which provides agentic AI capabilities that enable collaboration between developers and AI agents. Unlike standalone AI coding assistants that only help with new code creation, Duo Agent Platform is designed to work with developers across all tasks.

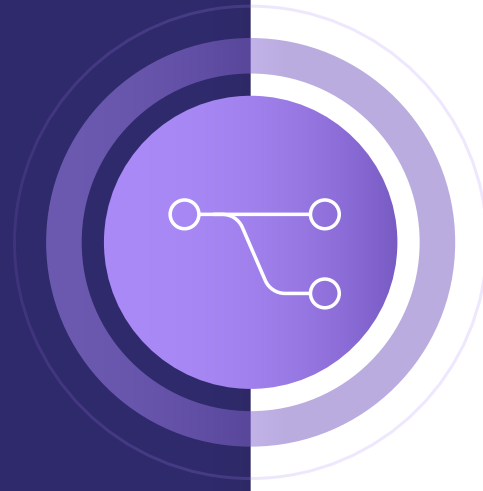
For Premium users, Duo has native access to your entire GitLab environment, providing it with the insights to address the broader workflow challenges that consume developer time.

Core AI capabilities (included with Premium)	Agent Platform capabilities (included in Premium public beta as of July 17, 2025)
Duo Chat helps teams understand unfamiliar code, improve existing code, and generate tests	Duo Agentic Chat transforms the chat experience with enhanced tools like chat history and slash commands for more efficient delegation
Duo Code Suggestions accelerates code creation and improvement by automatically generating lines of code	Software Development Flow streamlines complex development tasks by orchestrating agents across your codebase



The result

Your teams operate at peak velocity. They spend more time innovating and less time on mechanical tasks while maintaining higher code quality and a greater understanding of the entire development lifecycle. With agentic AI handling routine tasks across the full development process, developers can focus on the creative, technical work that drives business value.



/02/

Merge coordination breakdown

The problem: Uncontrolled code changes create chaos

When multiple developers work on the same codebase without mechanisms in place to thoroughly review and manage their code, their changes inevitably lead to stalled productivity, broken code, and customer-facing issues. Without proper merge controls, teams experience:



Poor code quality

Code changes bypass review processes, allowing bugs and security vulnerabilities to reach production.



Slow innovation

Developers spend more time resolving conflicts than writing new features.



Manual coordination

Manual merge management consumes valuable development time, forces teams to manually coordinate merges across regions and time zones.



Broken development flow

Teams can't maintain the fast, iterative cycles that modern software development requires.

One bad merge can trigger customer-facing issues that take days to fix and months to rebuild customer trust. Meanwhile, developers get frustrated with constant rework and conflict resolution, leading to decreased productivity and team morale.

The solution: Structured reviews and automated merge trains

Modern development teams need automation that prevents coordination problems before they start. GitLab Premium provides advanced CI/CD features that transform how teams handle code changes within their pipelines.

- **Merge request guardrails** provide quality control without slowing down development. This GitLab Premium feature allows teams to customize approval workflows with rules defining who must review code before merging, including options to prevent self-approvals and require authentication.
- **Merged results pipeline** tests the result of merging the source and target branches together. They're the difference between "my code works" and "my code works with everyone else's code."
- **Merge trains** eliminate the manual coordination that bogs down growing teams. Instead of developers constantly rebasing their work and manually managing conflicts, changes automatically queue for integration. Each change is tested alongside all other merge requests in the queue, confirming the compatibility of the requests before the final merge.
- **Duo Agentic Chat** can search your entire development environment, answering questions and supporting reviewers' understanding of bugs, readability, and coding standards.



The result

This automated approach transforms chaotic merge processes into predictable deployments. Distributed teams avoid waiting for someone in a different time zone to manually coordinate merges. Teams catch issues early when they're inexpensive to fix, and with the support of AI in every part of your workflow, tasks like identifying and resolving bugs become seamless.



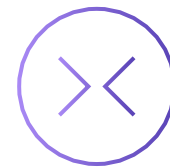
/03/

Development workflow blindspots

The problem: Teams can't optimize what they can't measure

Without clear metrics on where work gets stuck, how workload is distributed, or what issues are accumulating, you make decisions based on gut feelings rather than data. This problem gets worse when AI tools operate without understanding your full workflow.

AI might suggest code optimizations without knowing your deployment constraints, or recommend solutions that don't align with your team's established patterns. Altogether, teams end up in a cycle where the same problems persist because both your teams and AI don't have the insight needed to identify and address root causes.



Hidden process bottlenecks

Teams often don't realize when code reviews are taking too long, certain types of changes consistently cause delays, or specific workflow steps create friction.



Invisible workload imbalances

Development teams suffer from uneven workload distribution without realizing it, leading to burnout, missed deadlines, and suboptimal team performance.



Silent quality degradation

Code quality issues accumulate gradually, creating technical debt that eventually slows down development velocity.



Duplicative efforts

In larger organizations, different teams operate in isolation, making it challenging to share best practices, coordinate dependencies, or optimize workflows across the organization. Teams often duplicate efforts to solve the same problems.

The solution: Advanced analytics and reporting with AI-native insights

GitLab Premium provides visibility into development performance through integrated analytics, showing you where bottlenecks occur and where developers may need more training. The Duo Agent Platform takes the metrics a step further by helping you analyze and understand data from multiple sources. By asking AI agents questions about your analytics, you get a deeper understanding of the opportunities within your codebase.

Premium analytics capabilities

- **Productivity analytics** measure development velocity by tracking merge request completion times. These metrics help identify bottlenecks by showing which authors, labels, or milestones affect delivery speed.
- **Code quality reports** identify maintainability issues during code review. The pipeline details view shows all code quality findings that were found on the branch it was run on, providing teams with visibility into quality trends and enabling proactive quality management.
- **Contributor analytics** visualize individual contributions across your group by tracking team member activity over weekly, monthly, or quarterly periods.
- **Duo Agentic Chat analysis** reviews information from multiple sources to help you get more comprehensive answers. Use it to analyze pipeline logs and understand the status of issues or merge requests. For even more thorough answers, use [Model Context Protocol](#) to connect Agentic Chat to external data sources and tools.



The result

Teams can pinpoint exactly where work gets stuck, identify quality issues early, better balance workloads, and make data-driven improvements to their processes.



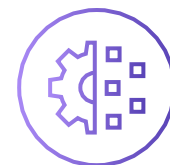


/04/

The planning paralysis

The problem: Project coordination becomes a juggling act

As development teams grow beyond a handful of people, coordinating work becomes exponentially more complex. Without proper project management tools integrated into your development workflow, teams struggle with fragmented planning, unclear dependencies, and bottlenecks that prevent efficient collaboration.



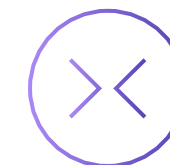
Fragmented planning across tools

Teams juggle separate platforms for project management and development, creating information silos and forcing constant context-switching between different systems.



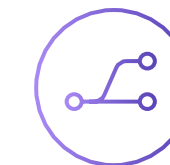
Limited visibility into large initiatives

Without epic and milestone management, teams lose track of dependencies and progress on complex projects that span multiple departments or timelines.



Single-person bottlenecks

When tasks can only be assigned to one individual, work stops whenever that person is overloaded, on vacation, or out sick, creating artificial constraints that slow delivery.



Inconsistent workflows

Different teams develop their own task management approaches, making cross-team coordination difficult and preventing consistent standards across the organization.

The solution: Integrated Agile project management

With GitLab Premium, teams can access enterprise-grade Agile tools within their DevSecOps platform. You can start by creating groups and projects, assigning team members roles, and determining their level of permission. These tools enable shared ownership and continued progress regardless of individual availability.

— **Milestones and epics** help teams plan large-scale initiatives across multiple projects to track dependencies, progress, and align on deliverables. This gives everyone clear visibility into the process and ensures that complex projects stay coordinated across different teams and timelines.

— **Advanced issue management** lets you dive deeper into each task with customizable workflows and multi-assignee capabilities. Teams can visualize project progress, dynamically adjust priorities, and collaborate on issue resolution without switching between different tools or platforms.



The result

Teams coordinate work seamlessly within their development environment, eliminating the overhead of managing separate project management tools while maintaining clear visibility into progress and roadblocks.



/05/

Troubleshooting without support

The problem: When things break, productivity comes to a halt

Every development team eventually faces the same scenario: a critical system fails, deployment pipelines break, or mysterious configuration issues bring work to a standstill. Without expert support, what should be a quick fix becomes an hours-long research project that derails your entire team.



The expertise gap

Most teams lack deep knowledge of every tool in their stack. When complex problems arise, developers must stop building features to become system administrators, troubleshooting through trial-and-error.



Cascading impact

System failures don't just affect one person — they sideline entire teams. When development infrastructure goes down, code reviews halt, deployments freeze, and multiple projects stall simultaneously.



Business continuity risks

Growing companies risk losing customers when unreliable systems damage their reputation. Enterprise organizations face SLA penalties, compliance violations, and significant revenue loss from extended downtime.

The solution: GitLab Premium support and high availability

GitLab Premium customers of both SaaS and self-managed instances have access to **Priority Support**, ensuring critical issues get immediate attention from platform experts. Instead of internal teams spending hours troubleshooting or waiting for responses to the GitLab Community Forum, you get both proactive reliability measures and reactive expert assistance when issues do arise.

If you're hesitant about making the transition, know that with priority support, you'll receive upgrade assistance in which a Support Engineer will review and provide feedback on your upgrade plan to ensure a smooth transition.



The result

Teams focus on building software instead of fighting infrastructure fires, with both preventive measures and expert help available exactly when you need it most.

Struggling with security roadblocks?

If security and compliance requirements are forcing your teams to slow down or take dangerous shortcuts, you may need GitLab Ultimate.

While GitLab Premium offers security features like Secret Detection and Audit Events, our top-tier plan provides advanced security capabilities. Built-in security testing, Security and Compliance Dashboards, and Vulnerability Management help you move faster while confidently maintaining security across your environments.

Discover if GitLab Ultimate is right for you >

Scale your development process: The cost of standing still

You've seen the five critical bottlenecks that limit development teams as they scale: merge coordination breakdown, manual development overhead, development blind spots, project coordination chaos, and troubleshooting without support. Each represents a choice between accepting limitations or investing in the tools that eliminate friction from your development process.

The hidden cost of free

Free tools provide an excellent foundation for teams starting their DevSecOps journey, offering core functionality that serves small teams and individual developers exceptionally well. However as teams grow, their needs become more sophisticated — whether due to growth, complexity, or operational requirements.

Continuing with free tools can create challenges that become more expensive as you grow.



Free challenges, expensive headaches

Developer productivity tax

Your team spends countless hours on repetitive tasks that AI could handle automatically, writing the same pipeline configurations repeatedly, and troubleshooting problems without expert guidance. This isn't just inefficiency — it's opportunity cost. Every hour spent on mechanical work is an hour not spent building features that drive business value.

Quality and security risks

Without advanced code review workflows, merge trains, and comprehensive quality reporting, teams accumulate technical debt and security vulnerabilities that become increasingly expensive to remediate. The cost of fixing problems in production exceeds the investment in preventing them during development.

Coordination overhead

Straightforward merge requests, basic pipelines, and informal communication work well for simple workflows. But as teams need to coordinate more complex processes, manage dependencies across multiple projects, or ensure consistent standards across distributed teams, your free plan's single-assignee limitations or basic workflow capabilities can create bottlenecks that slow down development.

Competitive disadvantage

While your team manually handles tasks that Premium users automate, your competitors move faster, ship more reliably, and maintain higher quality standards. In competitive markets, this operational advantage translates directly to business outcomes.

AI context gap

Your team uses AI tools that understand individual files but miss the bigger picture. Meanwhile, teams using integrated AI benefit from suggestions that consider their entire development context — security policies, deployment patterns, and team standards.



The GitLab Premium transformation

GitLab Premium removes the friction that prevents teams from reaching their full potential. With automated merge coordination, comprehensive analytics, expert support, and agentic AI that supports and accelerates the entire development workflow, teams operate at another level of efficiency and reliability.

The question isn't whether you can afford to upgrade to Premium — it's whether you can afford to continue operating with the bottlenecks that limit your team's potential. Every day spent architecting around free tools is a day your team could be shipping faster, building better software, and focusing on innovation instead of battling infrastructure problems.



Eliminate bottlenecks and transform development

Start your GitLab Premium plan today and experience what your team can accomplish when technology amplifies their capabilities.

Get started with GitLab >