



GitLab AI Guide For Enterprise Leaders:
Building the Right AI Approach for Your Software Innovation Strategy

A Future of Freedom, Flexibility, and Choice

Authors: Brian Wald, Global Field CTO, and
Emilio Salvador, VP of Strategy and Developer Relations



Table of Contents

Executive Summary	3
Introduction	4
Key Benefits of AI in the Enterprise	6
How to Implement AI for Long-Term Success	7
Use Cases: Doing AI the Right Way	9
Play the Long Game With AI	12
Appendix: AI Glossary	13

Executive Summary

What should enterprise leaders make of Generative AI in 2024 and beyond?

GenAI enables developers to bring immense computing power to nearly any engineering task by simply writing prompts in everyday language that result in accurate, useful coding solutions. The potential benefits to enterprises are immense. However, the implications of bringing AI tools into software development and deployment environments are just beginning to be fully understood.

We've talked with hundreds of GitLab customers, and some of their top questions remain unanswered: Can code written by AI be trusted? Will these tools deliver on promised increases in productivity and efficiency? Will developers use AI coding tools for intended scenarios, stretch beyond appropriate applications too quickly, or perhaps not use them? How might consumer-facing applications change as AI is integrated into them?

How your organization answers these questions can be the difference between a successful enterprise AI implementation, which can become a foundation for future innovation, and going back to the drawing board, which can lead to falling behind peers, competitors, and customer expectations.

Leaders must chart clear yet flexible paths forward and communicate rationales and roadmaps throughout their organizations.

One certainty is that AI tools are here to stay. Companies with software engineering organizations that want to recruit and retain top talent must introduce AI to the enterprise to compete. Continuous upskilling, adoption, and support are critical to safely, securely, and responsibly unlocking AI's power. So, too, is mapping out long-term AI strategies that ensure organizational freedom and choice as infrastructure and application needs shift, mature, and scale.

Leaders must chart clear yet flexible paths forward and communicate rationales and roadmaps throughout their organizations. That's why GitLab has created this guide for enterprise leaders. Our aim is to help you think about these issues across the whole geography of your organization and prepare your C-suite, executive leadership, and development teams for what AI can do today—and will do in the near future—to accelerate software development.

Introduction

The AI revolution is well underway. Transforming business. Presenting opportunity. Redefining the way we work.

Enterprises are no longer just experimenting with Generative AI; they are carefully integrating the technology into their core operations to drive efficiency, innovation, and competitive advantage.

This guide takes a strategic approach to AI adoption for DevSecOps teams. We'll explore some of the industry's best practices and equip you with practical steps to implement AI code assistants throughout your software development lifecycle, maximizing the positive impact of AI on your entire organization.

However, 51% of C-level executives feel their methods for measuring developer productivity are flawed or want to measure it but aren't sure how¹.

Executives acknowledge the importance of developer productivity when considering coding assistants, but measurement can be difficult. In a survey commissioned by GitLab, 55% of C-level executives agreed that developer productivity is important to the success of their organizations, while 57% of C-level executives agreed that measuring developer productivity is key to business growth.

However, 51% of C-level executives feel their methods for measuring developer productivity are flawed or want to measure it but aren't sure how.

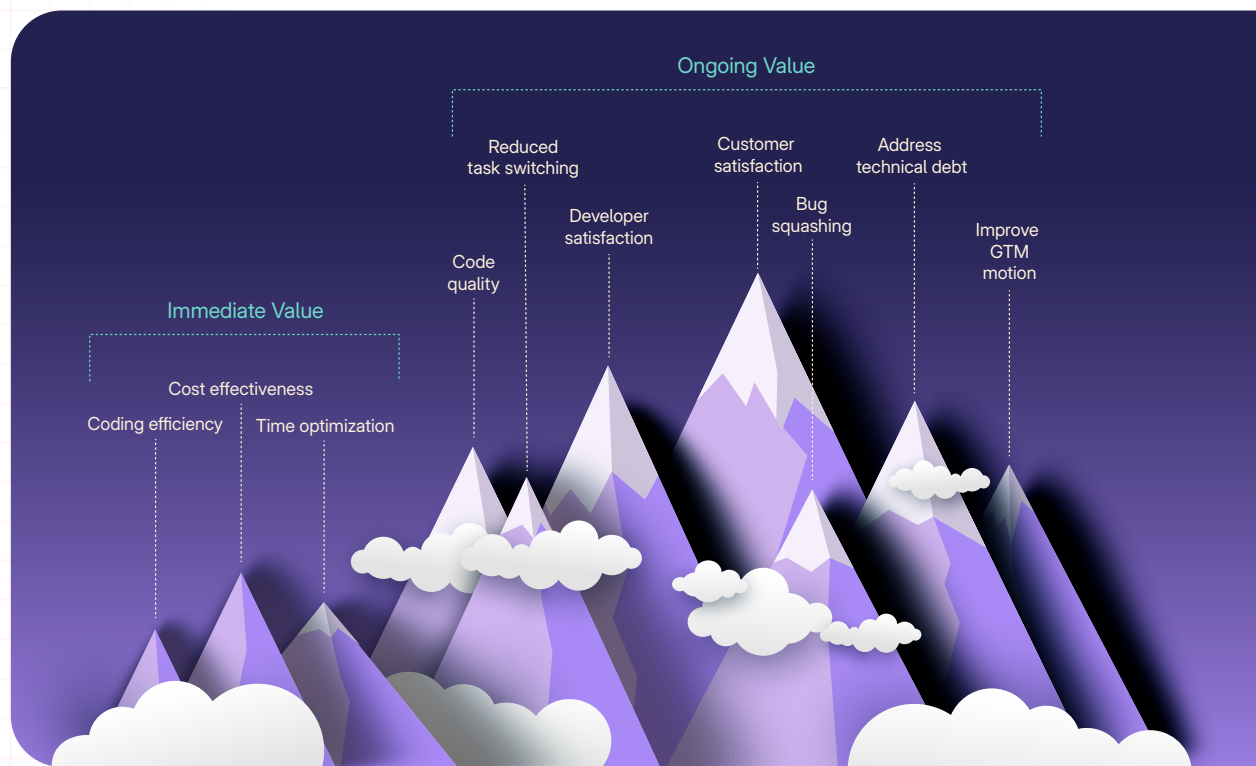
We'll explore what to expect over the next two years of AI evolution, providing insights and strategies for IT decision-makers to unlock AI's potential. Whether your company is an AI early adopter or just beginning its journey, GitLab is here to help you navigate these complex decisions and position your organization for long-term success.

¹ GitLab DevSecOps Report, 2024

The Experiment is Over: AI Integration is Essential

AI is a critical component of enterprise strategy, driving significant advancements across various sectors. Application development is not immune to this trend. According to a recent Gartner survey, 63% of organizations are currently piloting or have already deployed AI code assistants².

The initial experimental phase has passed, and organizations must now focus on integrating AI to deliver tangible results. This shift requires a strategic approach, ensuring that AI solutions align with business goals and provide qualitative and quantitative benefits. While the immediate value of code assistants is easily understood, the ongoing benefit to organizations is perhaps even greater and more lasting.



There's No Such Thing as Too Late for AI

Even if your organization has not fully embraced AI, the time to start is now.

By 2028, 75% of enterprise software engineers will use AI code assistants, **according to Gartner**, which defines them as tools that assist in generating and analyzing software code and configuration. And that is up from less than 10% in early 2023. Even though the adoption curve is steep, we are still relatively early in the hype cycle. In fact, if your developers haven't fully adopted an AI code assistant, they may be well-positioned to avoid some of the growing pains early adopters have experienced.

² Gartner Top Strategic Technology Trends in Software Engineering

Key Benefits of AI Code Assistants in the Enterprise

Building a robust AI strategy now can help you catch up to and even surpass your competitors. Here's how:

Speed Up Security Cycles

AI can significantly enhance your security posture by automating threat detection and response. AI-powered tools can identify vulnerabilities and potential attacks faster than traditional methods, helping to protect your systems and data more effectively.

Enhance Productivity and Resource Management

AI-driven tools, like GitLab Duo, can streamline development processes, improve team productivity, and optimize resource management. By automating routine tasks, AI allows developers to focus on more complex and value-adding activities.

For example, writing tests is often a time-consuming task that lags behind the code being deployed. AI can assist by generating tests in real-time, ensuring thorough coverage, and enhancing the overall quality of the code.

Optimize and Iterate with the Right Structure

A well-implemented AI framework that grants a holistic view of the software development life cycle can continuously improve itself as more data is collected and added to **context windows**. Enhancements in AI technology automatically integrate into your workflows, providing ongoing benefits without requiring significant manual updates.

Building a robust AI strategy now can help you catch up to and even surpass your competitors.

How to Implement AI for Long-Term Success

Implementing AI requires careful planning and consideration of the complexity involved and your current software development lifecycle. Where will AI create the most efficiency? In testing and deployment, maintenance, or some other area? Where might it create a bottleneck or add to a problem?

A thoughtful analysis of your current software development lifecycle, including tools and processes, can help inform where to implement a robust AI system that can deliver sustained value and how to iterate and expand over time.

Build a Data-Driven AI Structure

Creating a solid AI framework is crucial for long-term success. AI adoption requires modifying the entire software development lifecycle to centralize data. Centralizing data allows it to be used across the organization to inform prompts and allows RAG techniques to enhance generative AI outputs.

Data is the lifeblood of your organization's AI efforts. Feeding it into your AI systems will allow you to supercharge the output quality for both your developers and your customers and empower AI to help you measure success and identify areas for productivity optimization.

Data is the lifeblood of your organization's AI efforts.

Additionally, as you integrate AI into your applications, prepopulating them with appropriate training data will ensure they are production-ready from day one. Now is the time for developers and data scientists to collaborate on utilizing data warehouses and data lakes to facilitate access to training models and application usage.

Onboarding: Three Planning Considerations

Before starting your AI journey, craft plans and best practices and nominate a leader (or team) to review and recommend approaches specific to your organization's needs.

Create an AI Steering Committee With a Strong Leader

In some cases this may be a **Chief AI Officer** to oversee AI strategy and implementation. This doesn't have to be an immediate addition to the C-suite; it may be a transitional title that a VP assumes to coordinate AI usage across teams. But it should be someone empowered to look holistically at AI. The AI Steering Committee should bring legal, security, and engineering leaders together to build a structure for AI adoption. It's important to remember that AI success isn't possible without first addressing the privacy, security, and legal requirements your organization may face, and how AI adoption plays into continued compliance.

Establish Guardrails and Workflows

Establish guidelines and workflows to ensure AI is used responsibly and effectively. Specifically, set up automated testing, including using a security analyzer, to create a gating mechanism that ensures code is reviewed before being promoted to production. And beware of Shadow AI—the latest variation of Shadow IT—where workers subscribe to their own assistants while working on your code base, leading to IP leakage.

Prepare Now for Multi-Model Approaches

Utilize multiple AI models to cover a range of applications and improve robustness. One size does not fit all—LLMs tuned for specific tasks are proliferating. Enterprise systems are beginning to implement fine-grain controls to help DevSecOps pick the right LLM for the job—or automatically optimize the routing for a given task.

Quick Wins: Two Ways to Immediately Benefit from AI:

As your organization gets used to Generative AI tools, these two project templates offer examples of quick wins that will generate learnings to benefit almost any software organization.

Legacy System Modernization

Use AI to modernize and migrate away from legacy architectures and languages, enhancing capabilities in the process. Many organizations struggle to invest in legacy applications due to cost, timeline, and ROI. Typically, they are forced to choose a path of barebones investment, eschewing new features to focus on maintenance and security.

With AI, modernization is frequently not only cost-justified but achievable on a rapid timeline.

In particular, memory-safe programming languages offer an example of where AI can provide clear and immediate benefits. A [recent analysis](#) shows that memory safety is a leading cause of security vulnerabilities. Since AI can generate code in memory-safe languages today, governments and other organizations can have confidence that their refactoring efforts will comply with future regulations and trends toward memory-safe language adoption.

Developer Upskilling

Upskilling your developers to ensure they can use and maintain AI systems effectively is a task that can actually be accomplished with the help of Generative AI.

Training on new AI tools and methodologies is something developers can do almost instinctively by using GitLab Duo. Developers at all levels can learn effective prompt engineering techniques while maintaining coding skills by using [Code Suggestions](#) to learn from Generative AI as they solve problems. Junior developers, in particular, can benefit from an always-on coding partner to learn from, making them productive, efficient team members more rapidly than ever before.

Developers need toolchains they feel good about using—it's crucial to have their buy-in as their roles evolve from writing code to shaping how AI tools are used and what they generate. By focusing on architecture, quality, and performance, developers can use AI to drive greater organizational value.

Doing AI the Right Way

From Marketing Magic to Reality

To leverage AI effectively, cut through the hype. Sky-high, context-free promises like “55% faster coding” are still being made everywhere you look. Ask critical questions of your team and vendors to scope AI project requirements accurately, onboard AI solutions thoughtfully, and adopt best practices for sustained success.

Real Success

Here’s a stat grounded in a real scenario with a tangible impact: A leading financial services company has decreased pipeline outages by 90% since deploying GitLab in its enterprise, translating into hundreds of thousands of dollars in savings annually. This customer is also using the AI power of GitLab Duo, which has increased productivity for its DevSecOps team and the business.

Code Development: Companies using AI-driven code development tools reported faster release cycles and fewer bugs in production. Gartner says by 2028, systematic adoption of AI code assistants in 2023 will result in at least 36% compounded developer productivity growth³.

Give Developers a Voice and a Vote

AI’s role in democratizing technology is reshaping how we operate—and software developers are at the heart of this revolution. While many developers fear how AI may impact their careers, they also wield tremendous power in shaping how the revolution plays out. Their role is evolving from code creation to influencing the use of AI tools. Development team leaders who want to unlock the full power of AI should motivate their teams to play an active role in strategically integrating AI into their workflows.

AI’s role in democratizing technology is reshaping how we operate—and software developers are at the heart of this revolution.

³ How to Communicate the Value of AI Code Assistants, Gartner

Approach AI Responsibly

With new power comes great responsibility. Companies that don't use AI for software development risk being left behind. But leaders who rush into implementing AI without understanding its impact invite risks like security vulnerabilities, fines, customer attrition, and reputational damage.

Companies that don't use AI for software development risk being left behind. But leaders who rush into implementing AI without understanding its impact invite risks

Organizations must foster an environment where strategic discussions about AI are the norm, involving cross-team conversations with technical, legal, and AI service providers. An [AI Transparency Center](#) is a resource that can help decision-makers navigate these conversations and issues with clarity and confidence.

Go Beyond Output Metrics to Measure What Matters

Sometimes, with AI, the numbers are too good to be true. Efficiency gains in controlled, small-scale experiments are presented as achievable for large-scale enterprise deployments. Presenting these overly ambitious figures as targets is a recipe for failure. Focus instead on how AI can help generate new value for your organization.

With AI, success metrics can be redefined from raw statistics, such as lines of code produced, to business outcomes, such as faster software delivery timelines, developer happiness, and increased customer satisfaction scores. Good software development isn't about generating more lines of code faster; it's about efficiently solving problems and improving the quality of applications.

Integrating AI into the software development process frees developers' time, reduces cognitive load, and delivers greater value for organizations and end users.

You've Adopted AI—Now What?

Iterate, Iterate, Iterate

Implementing AI isn't like flipping a light switch. Developer teams need a trial-and-error period to determine how AI and other tools mesh with individual workflows. There could even be a short-term productivity decline before the organization realizes long-term gains. Leadership should prepare for this by emphasizing transparency and accountability throughout the cycles of development and iteration.

If you've worked through this guide to choose the right AI strategy for your organization, now is the time to have the courage of your convictions. Remember that no tool is ever immediately universally adopted nor deployed with instant maximum efficiency. Once you've prepared and trained your DevSecOps teams to use AI, give them time and space to learn how to make it work for your organization's unique use cases and needs, and work with other teams to establish best practices and new organizational models.

Engage the Full Impact of AI in your Organization

The past few years have seen massive shifts in how our society lives, works, and creates. Cloud adoption gave rise to digital transformation, which has now given way to the AI revolution. Companies and workers have had to adjust to significant shifts across every facet of their existence.

However, the opportunity in a revolution is evolution—providing a path to find new, more efficient ways to work and solve problems. Embracing AI with the same intentionality under which the shifts of the last few years have been navigated is the best way to ensure your organization is prepared and focused on building the enterprise of the future, today.

Cloud adoption gave rise to digital transformation, which has now given way to the AI revolution.

Conclusion: Play the Long Game

Think Ahead and Plan for Success

AI offers tremendous benefits, but success requires a holistic and strategic approach. Plan ahead for your DevSecOps teams by creating an environment where they can thrive with AI superpowers built into their tooling, allowing them to build great AI experiences for customers of your applications. Determine the appropriate leadership structure to support their efforts and show business value.

Map a Holistic Strategy

Treat AI integration like cloud transformation—mindful and deliberate. Avoid piecemeal solutions when possible and focus on a comprehensive approach that ensures sustainable growth and improvement.

Control and Optimize Consumption

Many AI vendors are moving from per-seat pricing to consumption-based billing. It will be important to manage this line item carefully—and it is likely that an AI tool can help. Manage consumption-based spending and avoid toolchain sprawl by ensuring AI investments improve outcomes effectively, using the measurement principles discussed above.

Let us Help

GitLab can help you bring AI into your organization responsibly, safely, impactfully, and effectively. This guide scratches the surface of our experience and understanding of working with organizations of all sizes on their AI use cases and needs.

Start a demo today, and see how we can be a partner in your organization's AI journey.

Appendix: Advanced AI Glossary

You already understand the basics. Here are terms you'll want to be conversant in as AI enterprise adoption grows in breadth and depth:

Adversarial Attacks

Strategies are used to trick AI models by providing deceptive input and exposing vulnerabilities that lead to incorrect predictions or classifications. Understanding and defending against these attacks is crucial for AI security.

Bayesian Optimization

A strategy for optimizing expensive-to-evaluate functions. It uses Bayesian statistics to model the function and select the most promising parameters to test, often applied in hyperparameter tuning and experimental design.

Contrastive Learning

A self-supervised learning approach that trains models by contrasting positive pairs (similar examples) against negative pairs (dissimilar examples). This technique improves the ability to learn representations that capture meaningful patterns in data.

Embeddings

A technique for representing high-dimensional data in lower-dimensional spaces while preserving meaningful relationships. It is commonly used in NLP to convert words into vectors that capture semantic similarities.

Explainable AI (XAI)

A set of techniques and methods that enable humans to understand, trust, and manage AI decisions. XAI focuses on making the decision-making process of AI systems transparent and interpretable.

Federated Learning

A decentralized approach to training machine learning models where data remains on local devices. Only model updates are shared, enhancing privacy and security by avoiding the need to centralize sensitive data.

Few-Shot Learning

An approach where models are trained to perform tasks with only a few examples. This mimics human learning capabilities and is particularly useful when data collection is challenging or costly.

Gradient Descent

An optimization algorithm minimizes the loss function in machine learning models by iteratively adjusting model parameters in the direction that reduces error. It is fundamental to training efficient models.

Hyperparameter Tuning

The process of optimizing the parameters that govern the learning process of machine learning models, such as learning rate and number of layers. Proper tuning can significantly enhance model performance and accuracy.

Knowledge Distillation

A technique where a smaller, simpler model (the “student”) is trained to replicate the behavior of a larger, more complex model (the “teacher”). This process helps in deploying efficient models with similar performance to larger ones.

Meta-Learning

Often referred to as “learning to learn,” meta-learning involves creating models that can adapt quickly to new tasks with minimal data. This approach is essential for developing AI systems capable of generalizing across different domains.

Neural Architecture Search (NAS)

An automated method for designing neural network architectures. NAS explores different network configurations to identify the most effective architecture for a given task, streamlining the model development process.

Retrieval-Augmented Generation (RAG)

A hybrid model combining retrieval-based and generative approaches. RAG retrieves relevant documents from a large corpus and uses them to generate more accurate and contextually appropriate responses or content.

Swarm Intelligence

A concept inspired by the collective behavior of decentralized, self-organized systems, such as ant colonies. In AI, swarm intelligence algorithms optimize problems by leveraging the collective behavior of simple agents.

Synthetic Data

Artificially generated data used to train machine learning models. It can augment real datasets, helping to overcome issues like data scarcity, privacy concerns, and bias.

Transfer Learning

A machine learning technique where a model trained on one task is adapted for a different but related task. This allows for faster training and improved performance, especially when data for the new task is limited.

Zero-Shot Learning

A machine learning paradigm in which a model can recognize objects or perform tasks it has never encountered before using semantic information or descriptions. This approach mimics human-like generalization capabilities.



GitLab

Software.
Faster.