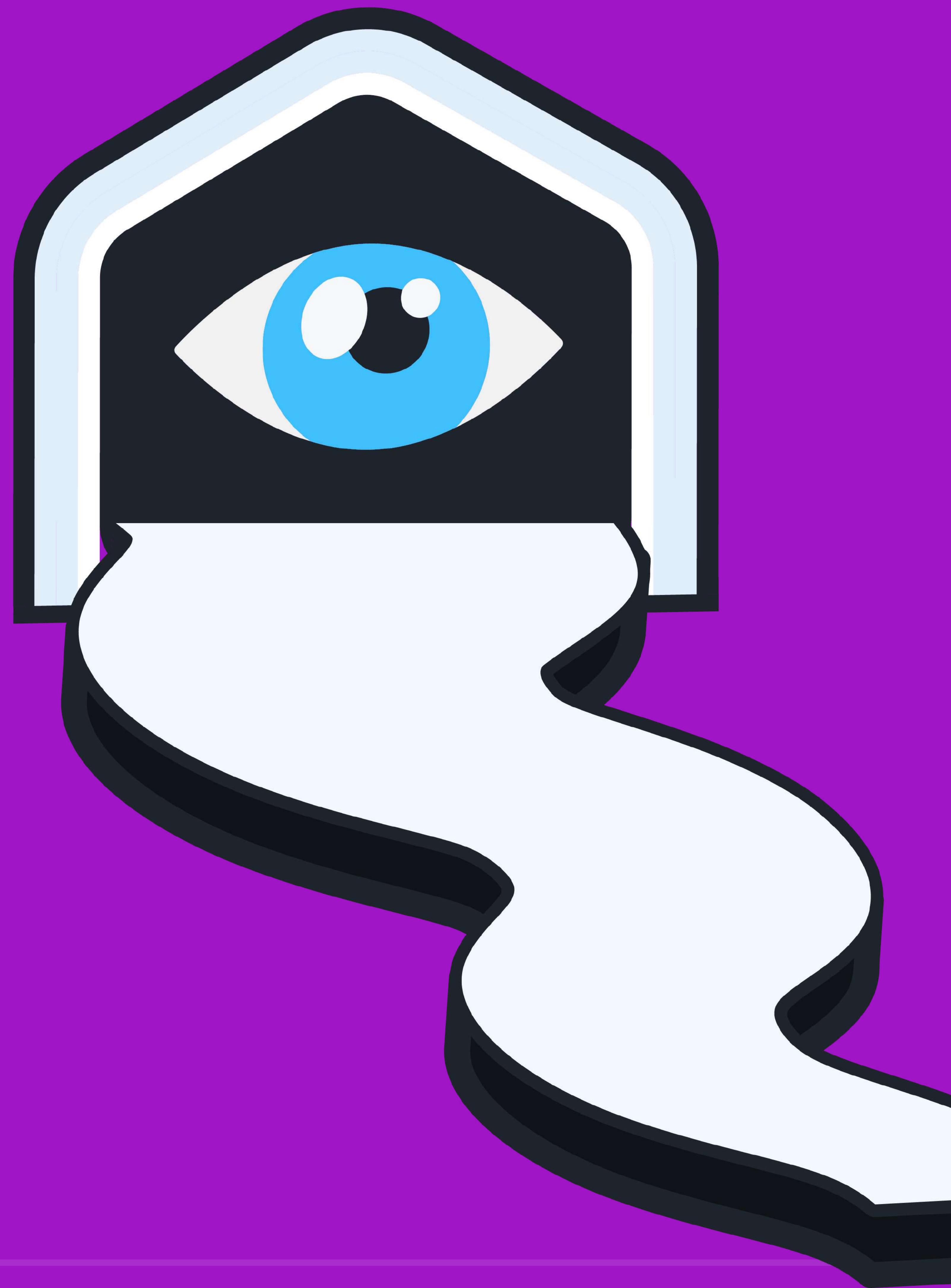


Tackling Tool Sprawl: A 5-Step Guide for Platform Engineering

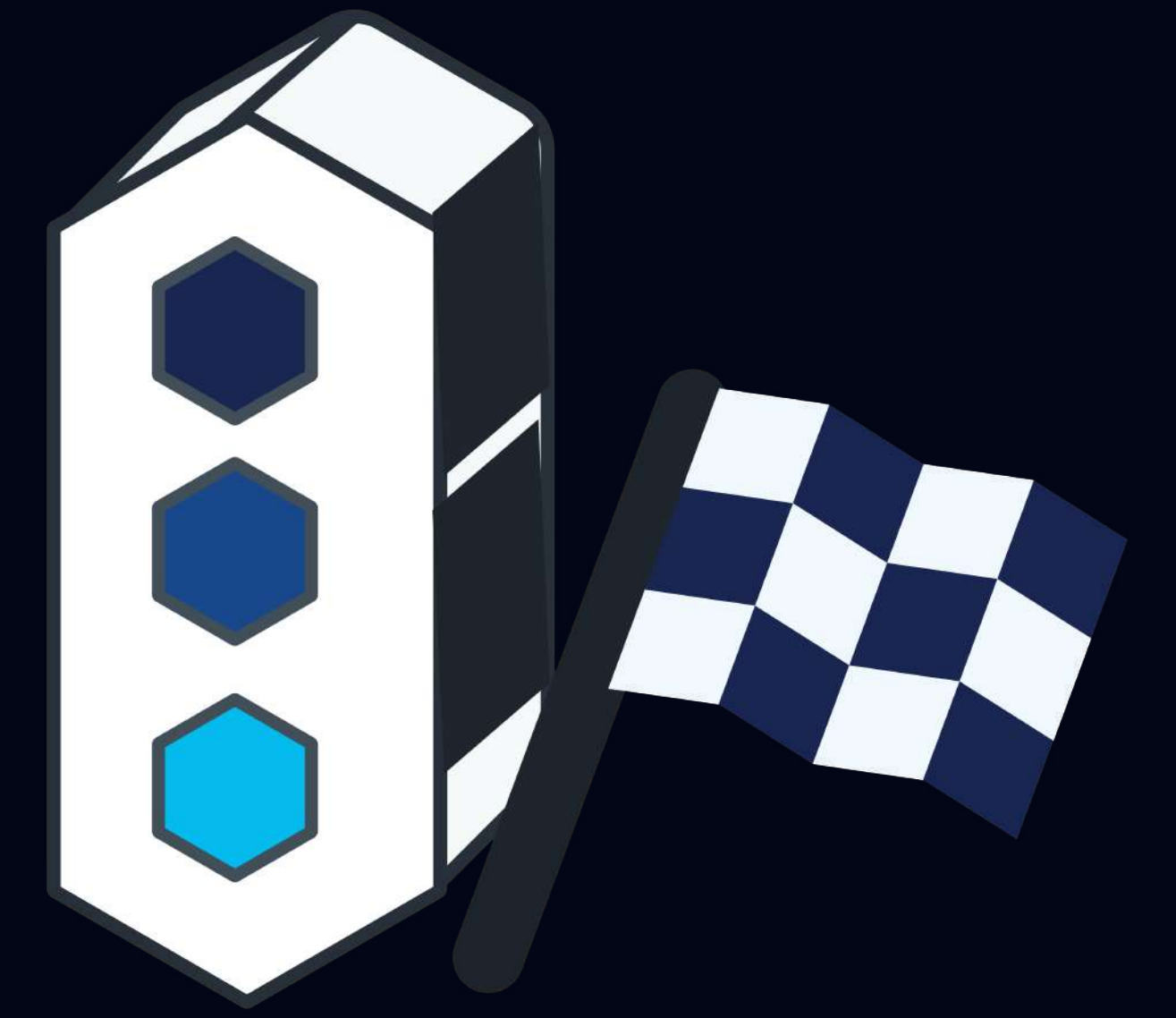
In the fast-paced world of software development, it's not uncommon for teams to utilize various point solutions to address specific challenges or accomplish certain tasks.

However, this approach can lead to a complex web of tools, often called 'tool sprawl,' which creates complications and hinders productivity.

Platform engineering now faces a new set of challenges. The sheer number of tools used in isolation across different teams can create inconsistent and incoherent internal processes. Furthermore, many point solutions fail to integrate seamlessly, exacerbating the tool sprawl problem and reducing organizational efficiency.



Addressing tool sprawl requires a careful and strategic approach and answers to the following crucial questions



1.

How is the data from various tools refined in a central location to get the insights needed to improve processes?

2.

How do we ensure processes are traceable so we can collect the evidence required to adhere to audit requests?

3.

How can we properly manage access, security and compliance across dependencies?

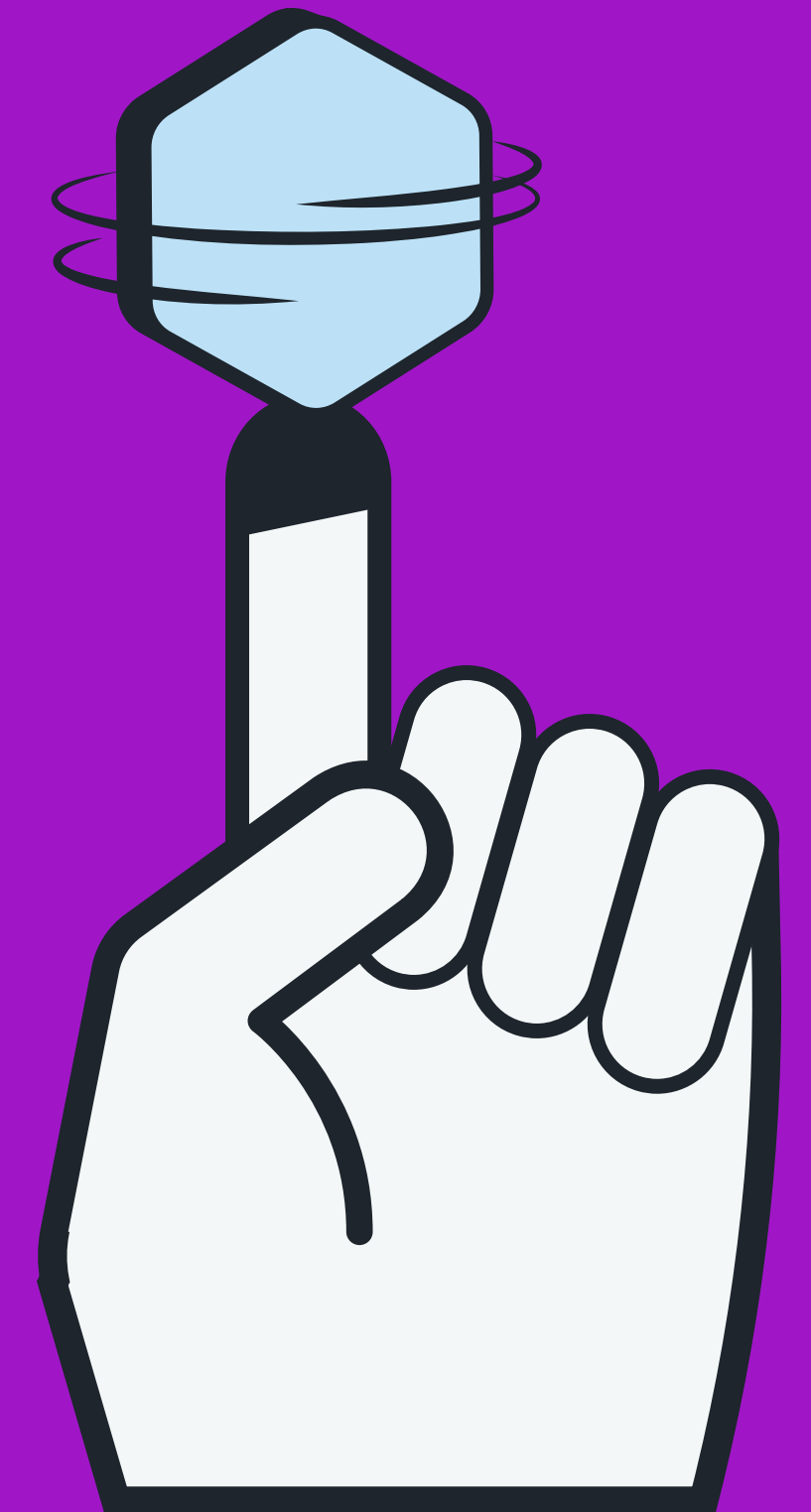
Providing an optimal developer experience is vital for creating exceptional products, often involving tool selection autonomy. However, managing numerous tools can distract developers from their core tasks and introduce compliance and governance issues. Put simply, developers are tired of toolchain management, with **66% stating they want to consolidate toolchains this year.**

This is where platform engineering comes into play.



5

Guidelines Platform Engineering Must Follow to Address Tool Sprawl



1. Map out your existing processes to identify and understand all the tools and people involved.

The first step to improving anything is to understand it fully. Optimization becomes feasible once platform engineering accurately grasps how components fit within value streams. Here are some strategies to gain this understanding:

- Conduct interviews with team members or send out an internal survey.
- Review existing documentation.
- Monitor processes from a central perspective.
- Develop flow charts, swimlane diagrams, or process maps.

These methods can assist in visualizing the stages of each process, who is involved, and what tools they are using to complete those steps.

2. Identify common patterns and create standardized blocks for easy reuse.

Mapping existing processes provides a clear view of the participants, tools, and flow. Now it's time to connect the dots by identifying common tasks within these processes, such as:

- Container image building
- Unit, system, or integration testing
- API, technical, or user manual documentation
- Deploying to Kubernetes
- Targeting cloud services like Lambda

By developing a reusable library of standardized blocks, teams can drive efficiencies. Team members can utilize this central resource for common tasks, optimizing their efforts.

5 Guidelines Platform Engineering Must Follow to Address Tool Sprawl



3. Find opportunities to trim out repetitive tools.

When evaluating your organization's toolchain, there are likely duplicative systems—for instance, using multiple artifact management solutions. In these cases, using standardized blocks abstracts the specifics of the underlying tools, freeing developers from knowing which tool they are using. Developers simply need a series of functional blocks for tasks such as pushing or pulling files or data.

This approach reduces the cost associated with licensing and simplifies the software development process. Developers no longer have to learn and juggle multiple tools that serve the same purpose, which is critical to ensuring an optimal developer experience. This also makes managing these tools easier for the platform engineering team.

The selection of tools becomes easier to manage, provided the platform engineering team has insight into where and how each tool is used. Moreover, removing any unnecessary elements contributes to a more orderly system.

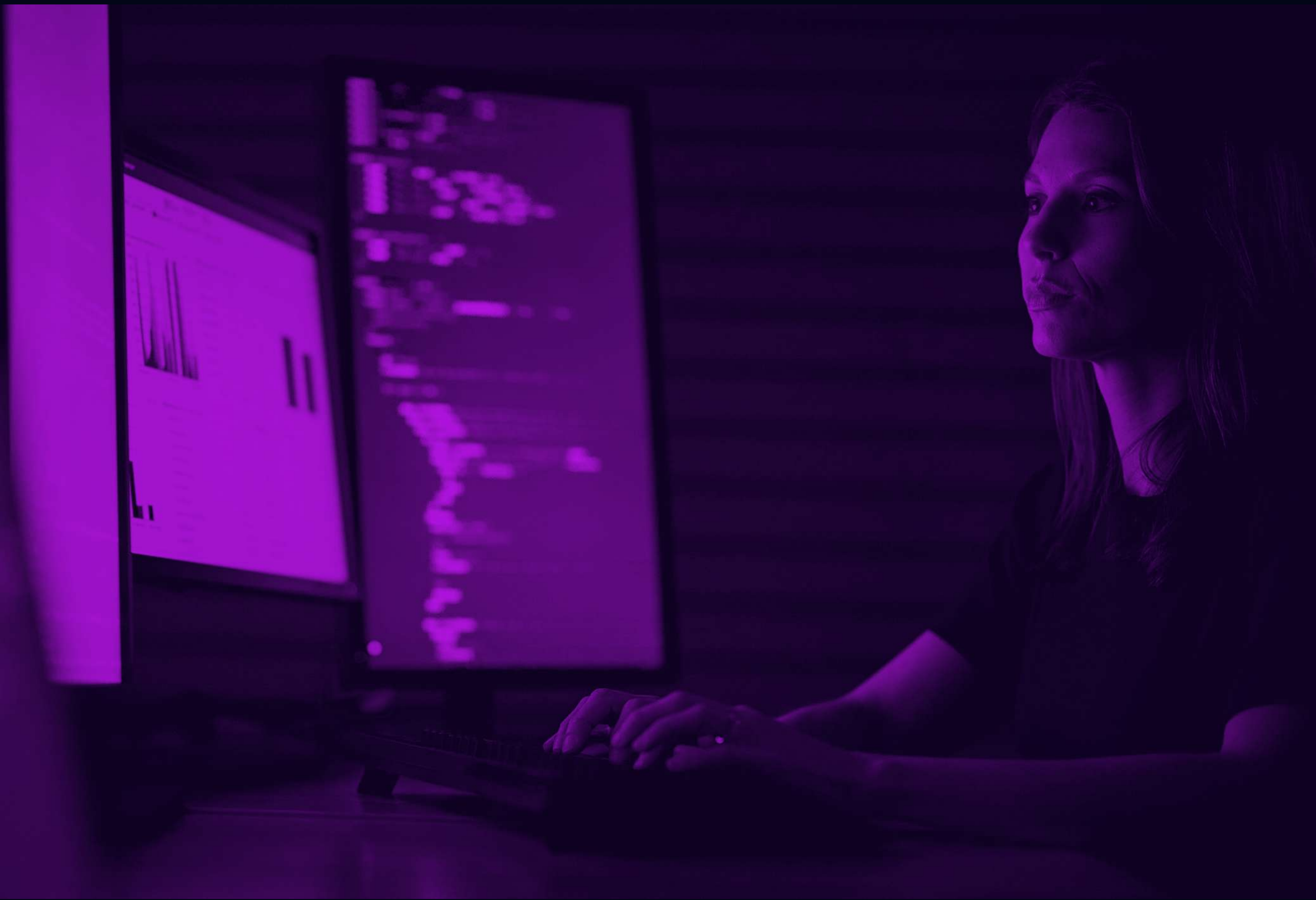
4. List additional steps the organization requires and ensure blocks are available.

While it's crucial to streamline processes, don't compromise on essential security and compliance checks. Removing these could lead to substantial problems.

Platform engineering teams, not developers, should establish security and compliance procedures. They can do this by establishing crucial blocks for security and compliance. This way, with parameters pre-defined, developers can confidently deploy secure, compliant code using their preferred tools to optimize their experience.

Maintaining software security and compliance will further boost productivity and ensure smooth audits and successful software delivery.

5 Guidelines Platform Engineering Must Follow to Address Tool Sprawl



5. Create golden paths based on these blocks that development teams can quickly adopt.

Golden paths are well-defined, task-oriented paths to facilitate predictable, consistent software development. They explicitly outline the tools, processes, and guidelines for testing and delivering software.

Platform engineering should create golden paths to integrate the prior four steps. This furthers consolidation of relevant blocks into a standardized process while still allowing development teams the flexibility to incorporate their blocks as needed.

Golden paths empower developers to use their favorite tools and provide a structured framework for managing the complexity of maintaining multiple toolchains with numerous different tools.

CloudBees Platform

CloudBees provides platform engineering with the necessary controls, visibility, and traceability to deliver effective, compliant software. With CloudBees, development teams can stick to their routines while platform engineering retains the control and functionality needed for smooth operations.

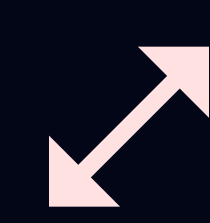
CloudBees offers an open and extensible platform to streamline all software development lifecycle (SDLC) efforts, fostering golden paths construction while maintaining a secure, stable platform.



The CloudBees platform enables



Increased productivity: Developers can rely on the platform to take care of the software delivery process, letting them stay focused on their code.



Enterprise scale: Regardless of your utilization growth, the platform is designed to support you.



Cross-functional collaboration: The platform helps identify and integrate best practices into reusable 'golden paths' for development teams.



End-to-end security and compliance: Automated security and compliance checks are integrated into workflows. By abstracting these from the pipeline, developers remain focused on writing code while providing risk-based evidence to satisfy auditors.



CloudBees provides the leading software delivery platform for enterprises, enabling them to continuously innovate in a world powered by the digital experience.

CloudBees enables organizations with highly complex environments to deliver scalable, compliant, governed, and secure software from the code a developer writes to those who use it. The platform connects with other best-of-breed tools, improves the developer experience, and enables organizations to continuously bring digital innovation to life to unlock business outcomes that create market leaders and disruptors.



Join Our Waitlist

cloudbees.com/products/saas-platform

Be the first to learn more about how the CloudBees platform can help your organization optimize productivity despite tool sprawl.