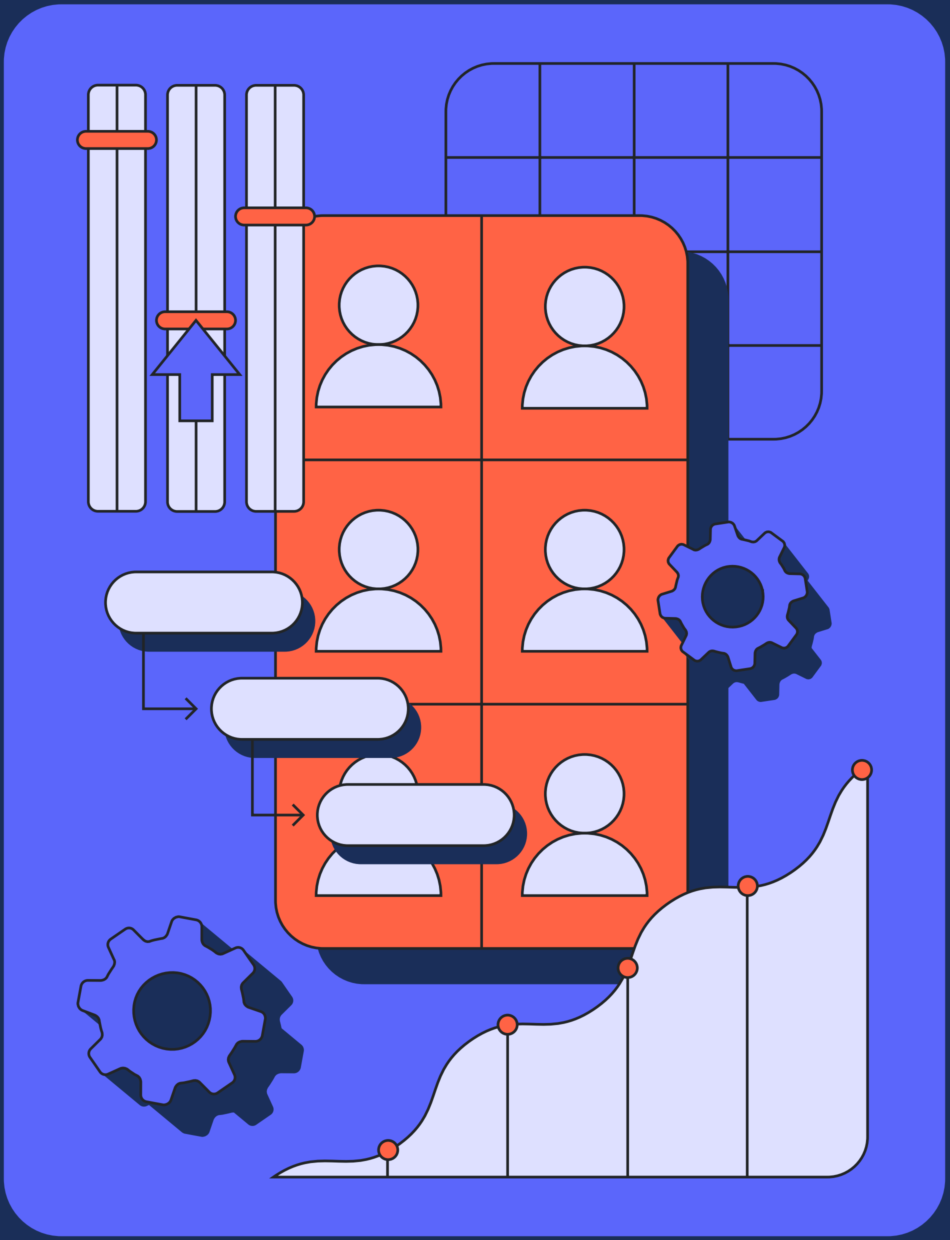
Managing Bot-Generated PRs & Reducing Team Workload by 6%

How dependency management bots are affecting your Developer Experience and Productivity.



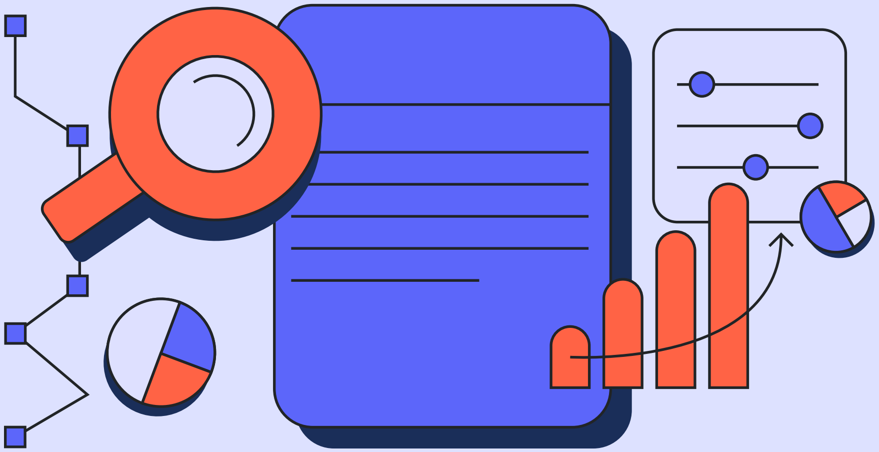
Data Report

LinearB

Table of Contents

The Bots Are ~~Coming~~ Here

One evening, a pair of data scientists at LinearB were preparing an insights report for a LinearB enterprise customer, when they found something unique in the data: a series of pull requests (PRs) without any human activity. When they looked closer, our scientists found that more than 10% of the workflow data in their analysis consisted of bot-generated PRs.



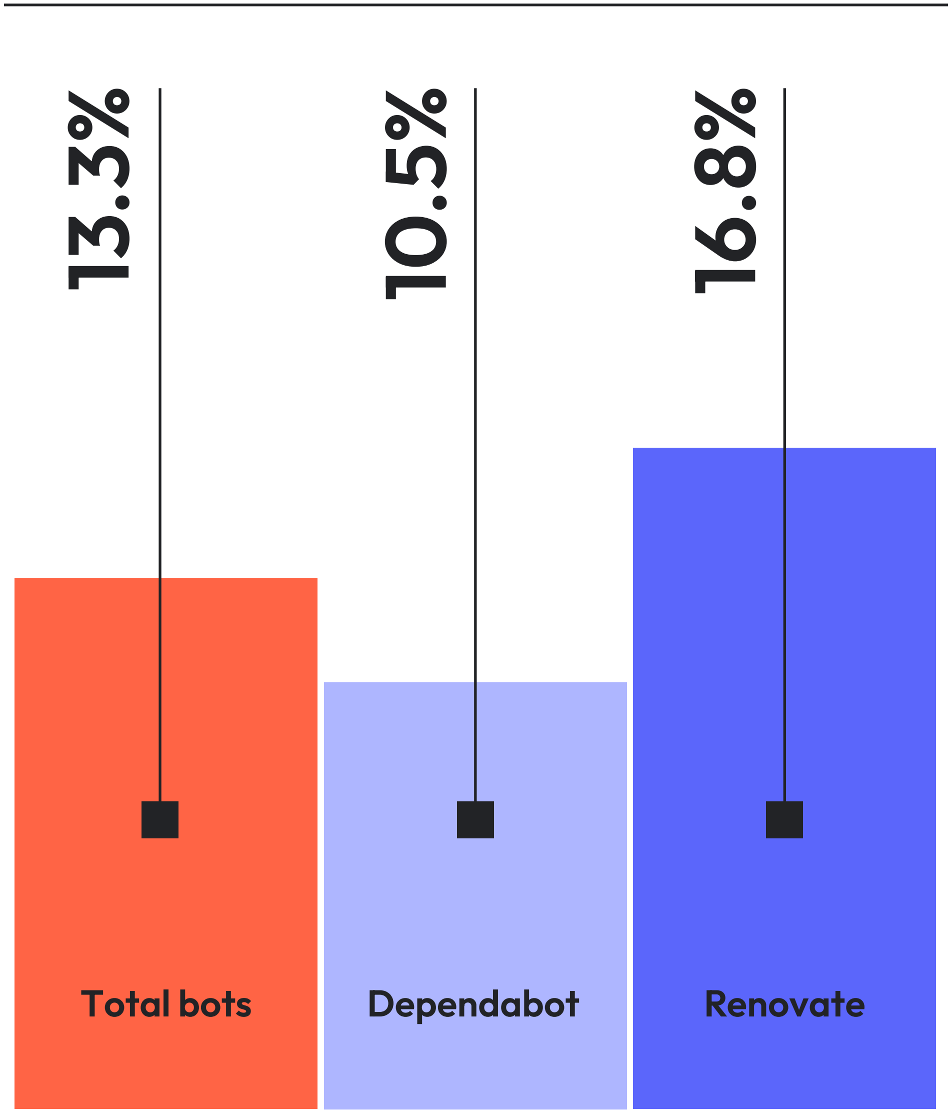
The next day, they brought this finding to the engineers at this Fortune 500 Company, who were surprised to learn that such a large portion of their submitted PRs were bot-generated (via Dependabot). Of course their use of a common dependency management tool wasn't a revelation, but the sheer number of pull requests, and the unique load this placed on their processes – for instance, these PRs were unaccounted for in Jira – compelled them to take a step back and reevaluate their bot management processes.

This revelation inspired us to dig deeper into a wider pool of anonymized data, and the following LinearB research report was born.

Our research revealed that bot-generated PRs make up 13.3% of the total number of code submissions at software engineering organizations that use tools like Renovate or Dependabot.

Key Takeaway

Depending on your engineering org size, this translates to thousands, or even tens of thousands, of bot-created PRs per year.



Bot-Generated PR Density by Vendor

Teams with more than 100 bot-generated PRs in the last 6 months.

Introduction

So what happens to all of these PRs? In most cases, they form an orphan workflow, with no representation in the project management tools. Reviewing these pull requests creates a huge load on teams. Ignoring them, besides cluttering up the git instance, defeats the purpose of using dependency management tools, exposing your codebase to vulnerabilities and lapses in compliance.

With more than 1 of every 8 PRs generated by bots today, a ratio likely to skyrocket with the imminent introduction of GenAI PR agents, we recommend engineering teams manage these PRs with a system that provides visibility, ownership and automation against this growing source of changes to your codebase. Our data shows that organizations who take on this initiative can reduce this additional load by over 40%, while also making drastic improvements in their security and compliance posture.

With data sourced from 2,800+ dev teams and over 3 million PRs, the 2024 Bot Automation Research Report breaks down:

1 **Research Insights**

The state of bot usage today and where bot PRs can be automated.

2 **How To Measure & Manage Your Bot Generated PRs**

Leveraging SEI (Software Engineering Intelligence) automation to identify and auto-approve over 40% of dependency update PRs.

3 **The Future of AI Agents**

Where the industry is headed regarding advancements in AI agents and bot-generated code.

Research Insights

Insight No. 1

13.3% of all PRs* are bot-created today.

Insight No. 2

96.2% of all bot-created PRs* are not linked to a PM tool.

Insight No. 3

Teams using SEI automation can safely auto-approve up to 84% of their bot PRs (patch and minor updates) and auto-merge up to 41.2% (patches only), resulting in an average reduction of 6% in the PR reviews required from the dev teams.

Insight No. 4

54% of Dependabot updates (major, minor, & patch) are deleted without action.

Insight No. 5

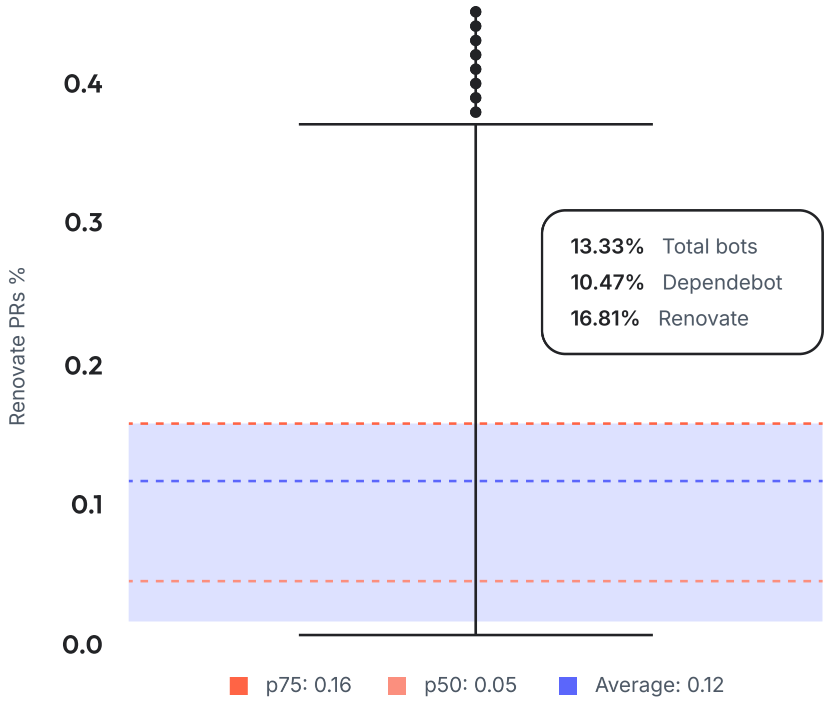
The average vulnerability time for dependency PRs is 12.27 days.

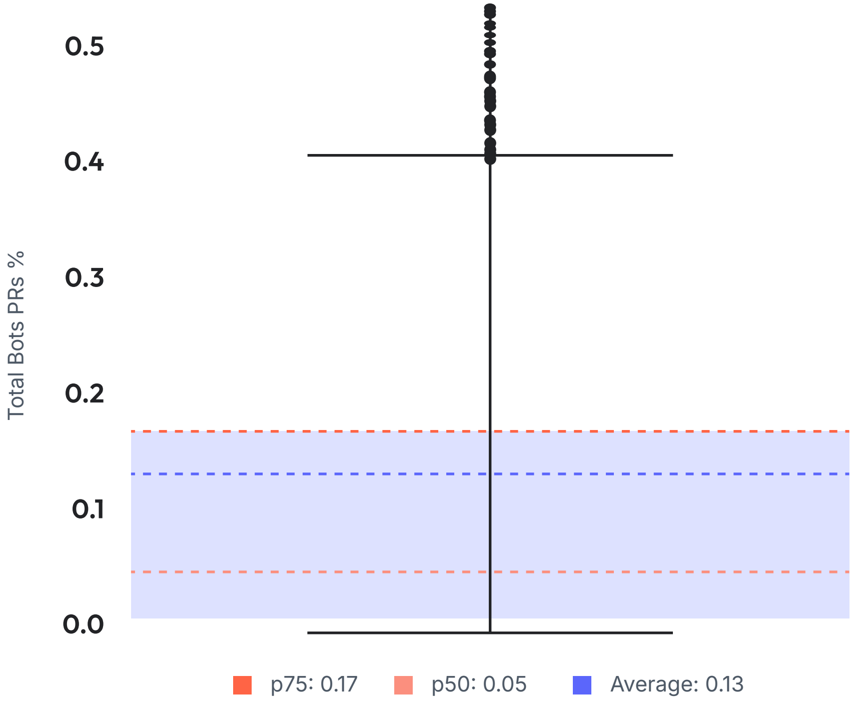
* From a sample of 3,098,246 PRs analyzed in this report.

1 Insights

Of teams currently using Renovate or Dependabot, 13.3% (average) of all their PRs* are bot-created. Renovate creates, on average, 16.8% of PRs, and Dependabot creates 10.5%.

* From a sample of 3,098,246 PRs analyzed in this report.





Renovate Bot-Generated PR Density

Criteria: Teams with more than 100 bot-generated PRs in the last 6 months.

Total Bot-Generated PR Density

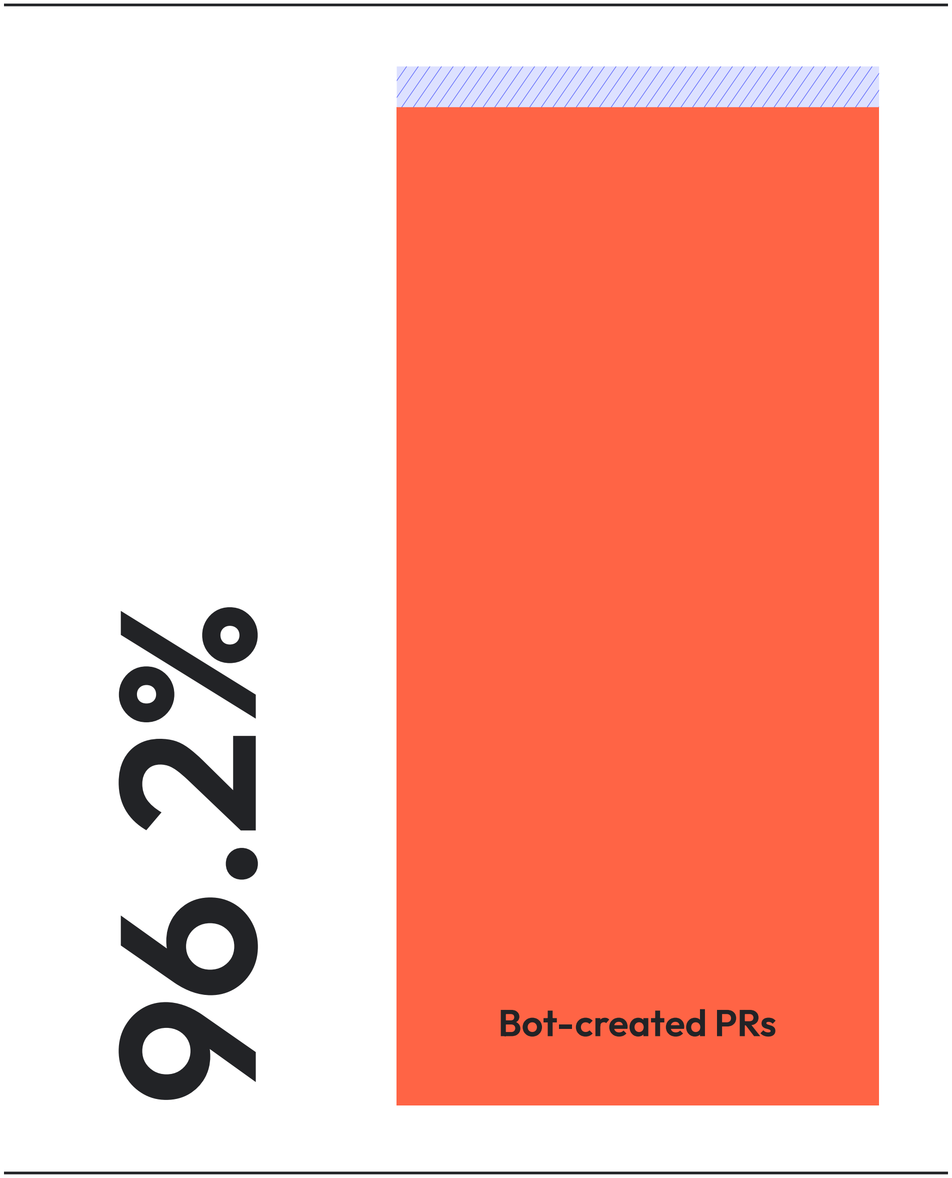
Criteria: Teams with more than 100 bot-generated PRs in the last 6 months.

2 Insights

96.2% of all bot-created PRs are not linked to a PM tool (96.6% for Dependabot and 98.7% for Renovate).

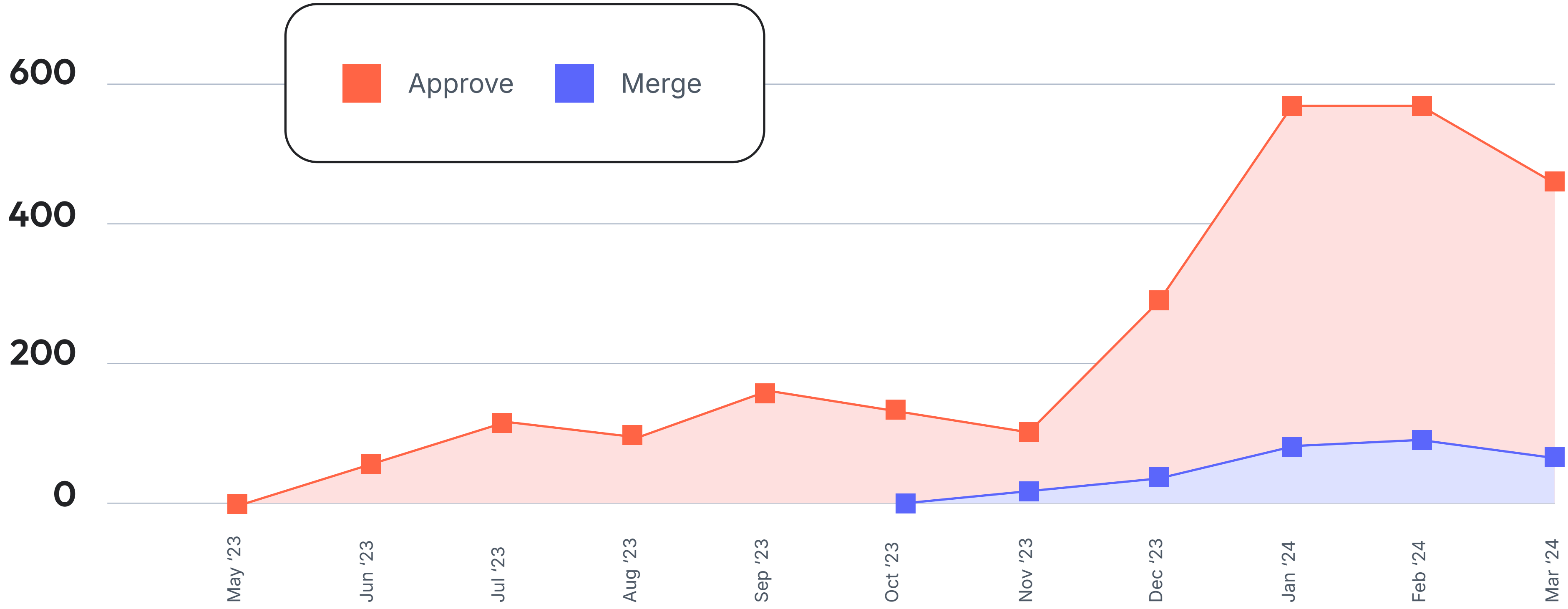
Key Takeaway

This presents a major risk to JIRA hygiene, untraceable work, and compliance (e.g. SOC 2, which typically requires that every PR be linked to a project management ticket).



Percentage of Untraceable Bot-Created PRs

Fortune 500 Company Uses LinearB to Auto-Review 3,000+ PRs Annually



Dependabot Automation Adoption Over Time

> “In the past 7 months, we’ve managed to significantly improve our workflows using SEI automation from LinearB (gitStream).
>
> Now, we auto-approve PRs for specific files like READMEs, images or code formatting – this frees us up for more complex reviews.”

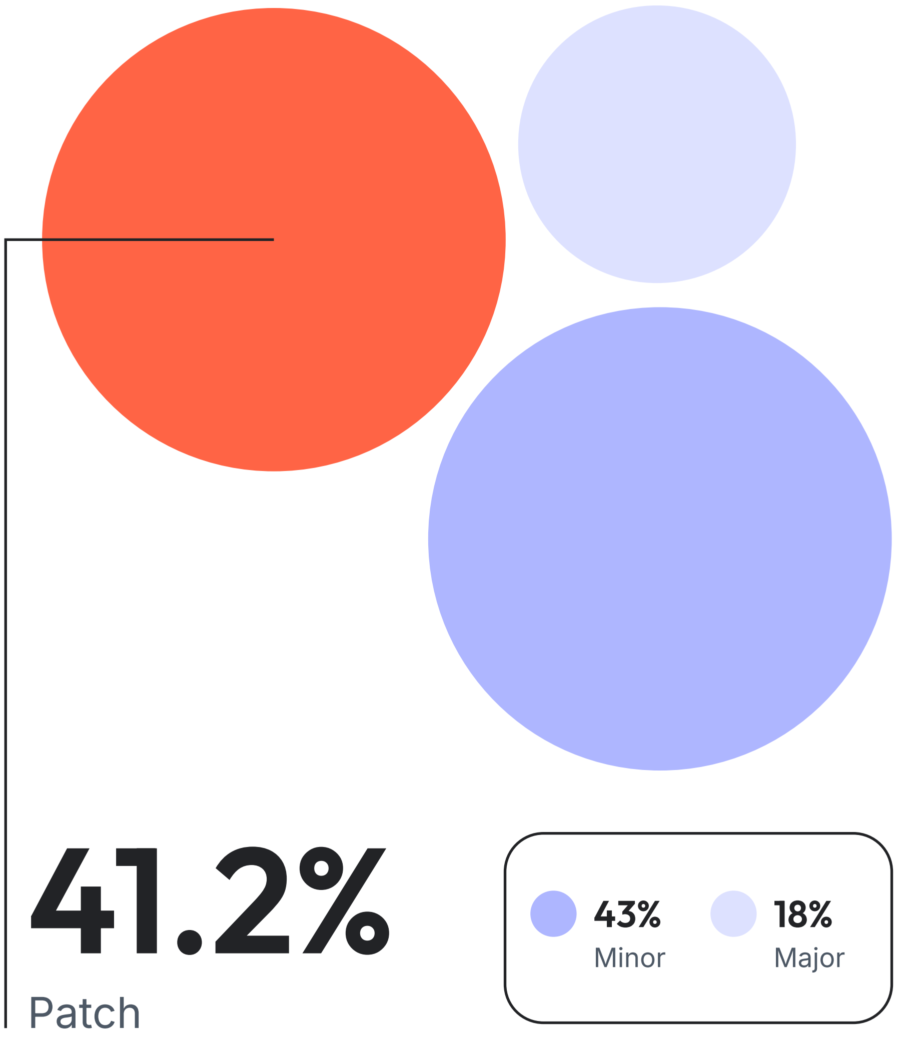
Engineering Manager

Fortune 500 Company

3 Insights

The data revealed that 41.3% of all Dependabot PRs are patch updates (which are often rubber-stamp approved, and can typically be safely auto-merged) and 42.96% are minor updates (which can be auto-approved).

In total, SEI automation can safely auto-approve up to 84% of your bot PRs (patch and minor updates) and auto-merge 41.15% (patches only).



Average Distribution of Version Bumps

Note: This data is from Dependabot users only.

As a quick refresher, updates to the library dependencies are broken down into the following three categories:

Patch

Patch updates typically address bug fixes and security vulnerabilities. They do not introduce new features or functionality, and are, generally, backward-compatible. For most teams, Bot-authored patch updates can be auto-merged.

Minor

Minor updates usually introduce new features or enhancements to existing functionality. They may also include bug fixes, but they tend to focus primarily on new capabilities. In most cases, bot-created minor updates are backward-compatible and can be auto-approved.

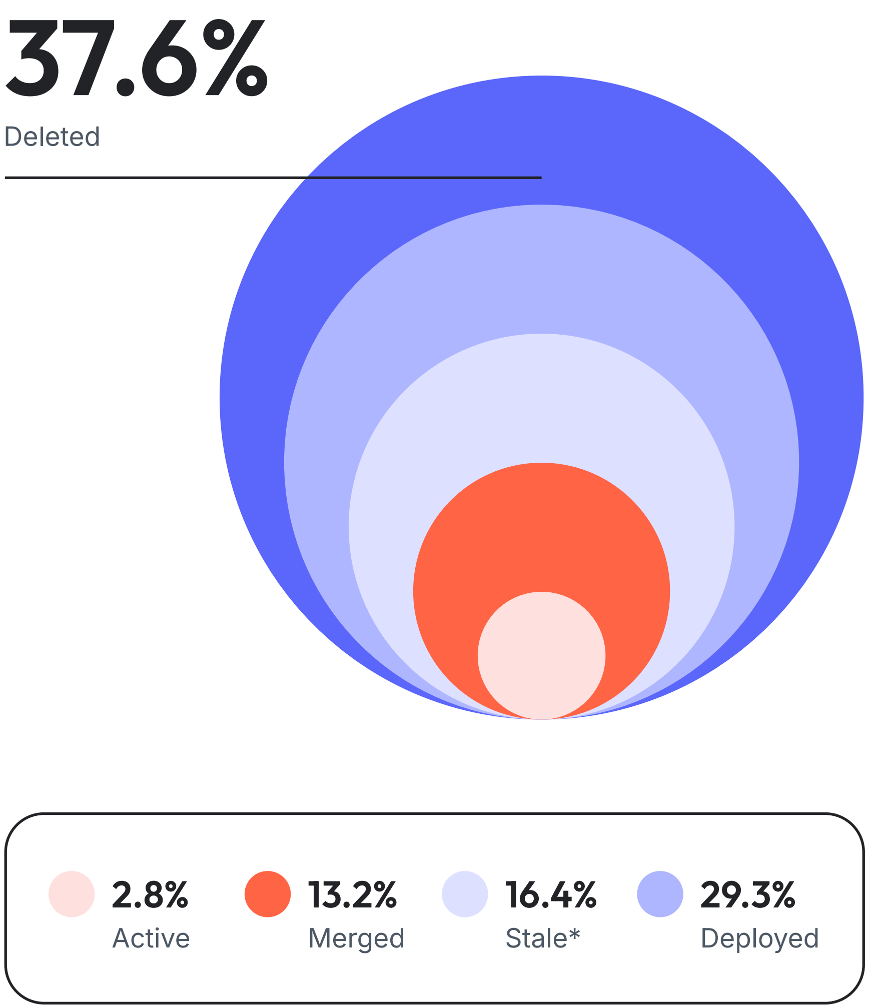
Major

Major updates represent significant milestones in the software's development. They include new features, enhancements, and in some cases, substantial changes to existing functionality. They often introduce breaking changes, meaning existing functionality or APIs might not work as before without modifications. Generally, bot-authored major updates need to be thoroughly reviewed and tested before they are merged.

4 Insights

A large portion of these these bot-authored PRs are ignored and eventually deleted – no doubt due to their huge volume. Over 37.5% of all Dependabot updates (major, minor & patch updates) are deleted and never acted upon. In addition, some 16.4% of these PRs are stale, likely to be deleted at some point in the future.

This dynamic hints at the toil required to handle these PRs. Even just cleaning them up is work developers shy away from. But the deeper impact of ignoring or stalling these PRs is in the longer periods where the codebase has deprecated or vulnerable dependencies, defeating the entire purpose of the dependency bots.



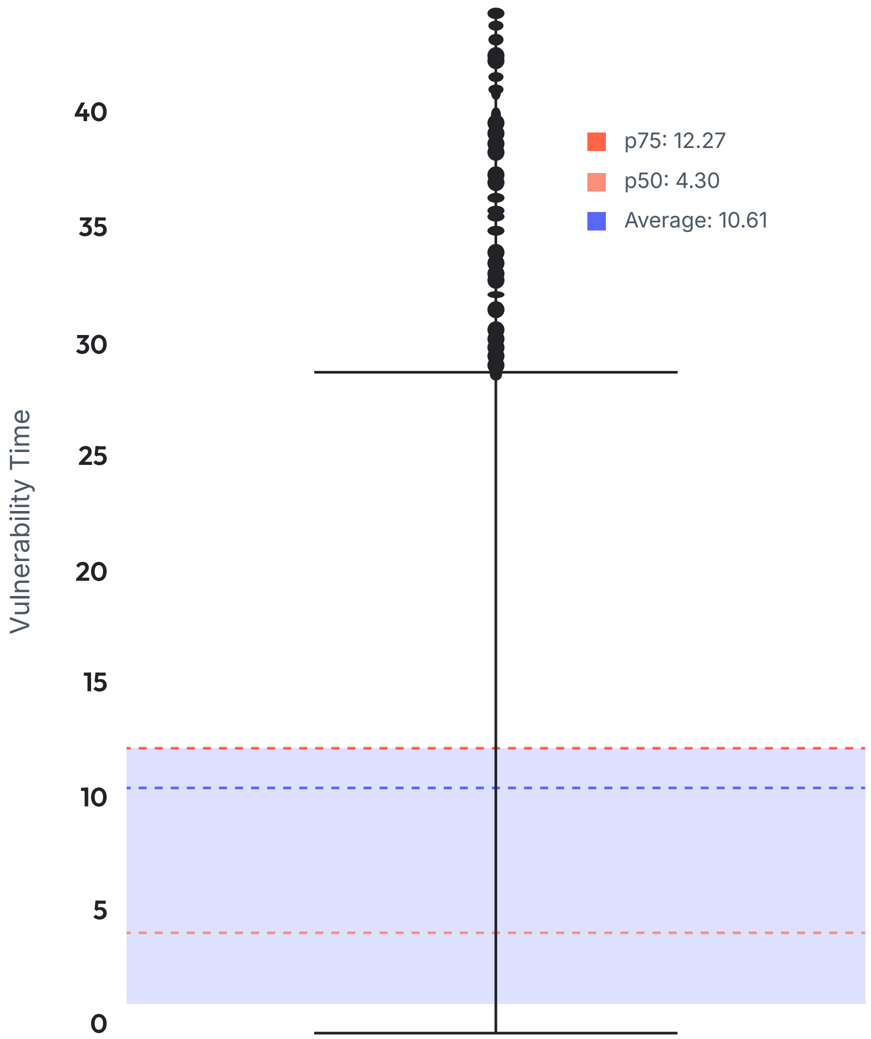
Branch State Version Bumps Distribution

* Stale defined as 7 days without Git activity.

5 Insights

We define vulnerability time as the average time it takes for a package update PR to get merged to the codebase (i.e. the average time your repo is using old, deprecated library versions and is possibly vulnerable to security flaws).

In our analysis, we found that the typical vulnerability time for bot-created PRs is 12.27 days. This is a sizable delay that can be expedited by automating bot-authored dependency PRs, especially patch updates which are the common form of delivery for security updates



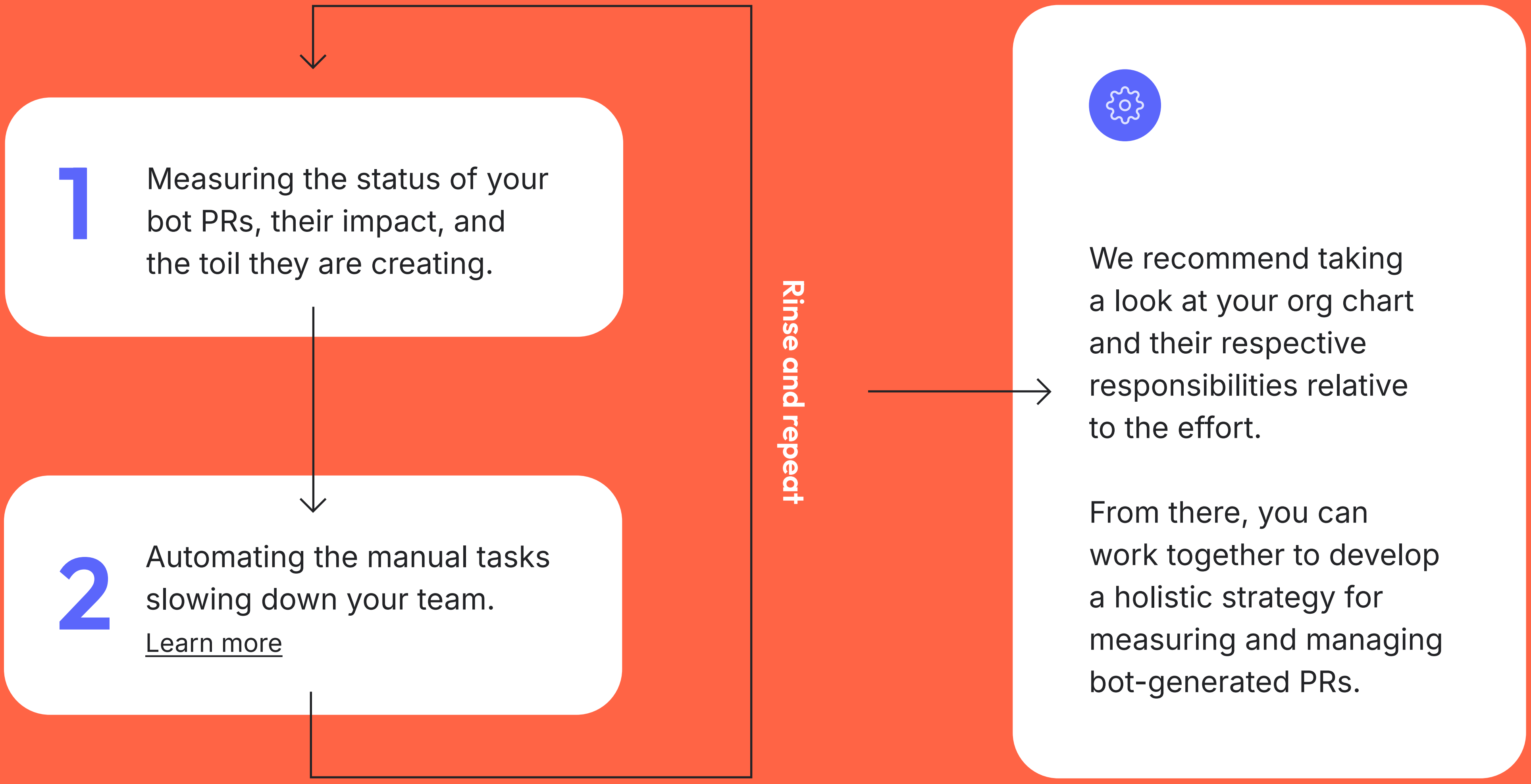
Bot-Generated PR Vulnerability Time

Starting a Bot Initiative

Every member of an engineering organization is responsible for the security and compliance of their organization's codebase. While this has become an intrinsic truth in modern software engineering, it has also created a certain amount of toil for developers. As you can read in the final chapter of this report, dependency bots are just the tip of the iceberg.

Realizing this, we recommend that engineering organizations start a Bot Initiative, with representation of both Engineering and Security, covering dependency bots as well as any other automated flows that create or manipulate pull requests. The goal of this initiative is to maximize the benefits of PR bots, while minimizing developer toil.

This initiative has two functions run in lockstep:



1 Measuring the status of your bot PRs, their impact, and the toil they are creating.

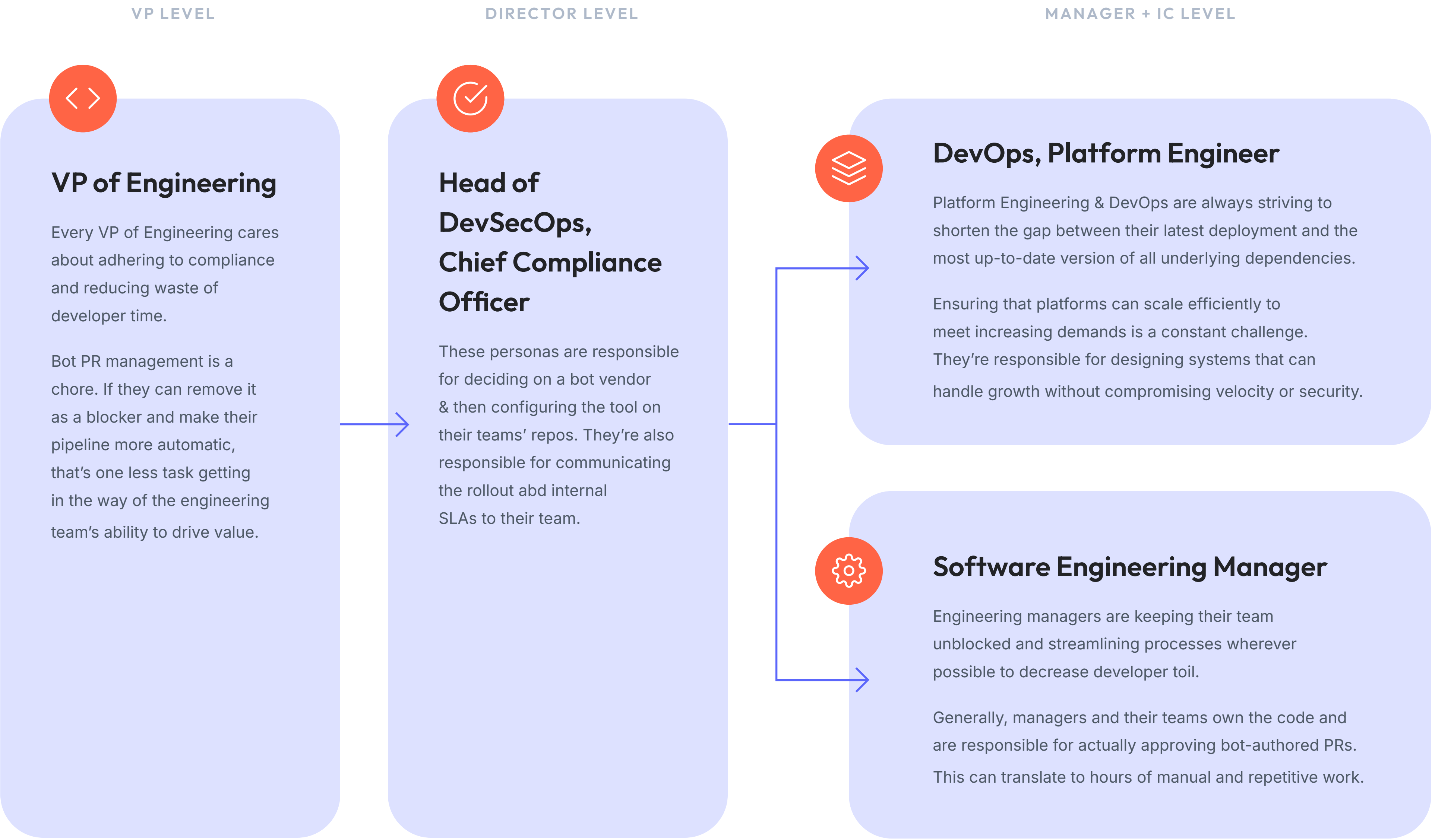
2 Automating the manual tasks slowing down your team.
Learn more

Rinse and repeat

We recommend taking a look at your org chart and their respective responsibilities relative to the effort.

From there, you can work together to develop a holistic strategy for measuring and managing bot-generated PRs.

Persona Flowchart



VP LEVEL

DIRECTOR LEVEL

MANAGER + IC LEVEL

VP of Engineering

Every VP of Engineering cares about adhering to compliance and reducing waste of developer time.

Bot PR management is a chore. If they can remove it as a blocker and make their pipeline more automatic, that's one less task getting in the way of the engineering team's ability to drive value.

Head of DevSecOps, Chief Compliance Officer

These personas are responsible for deciding on a bot vendor & then configuring the tool on their teams' repos. They're also responsible for communicating the rollout abd internal SLAs to their team.

DevOps, Platform Engineer

Platform Engineering & DevOps are always striving to shorten the gap between their latest deployment and the most up-to-date version of all underlying dependencies.

Ensuring that platforms can scale efficiently to meet increasing demands is a constant challenge. They're responsible for designing systems that can handle growth without compromising velocity or security.

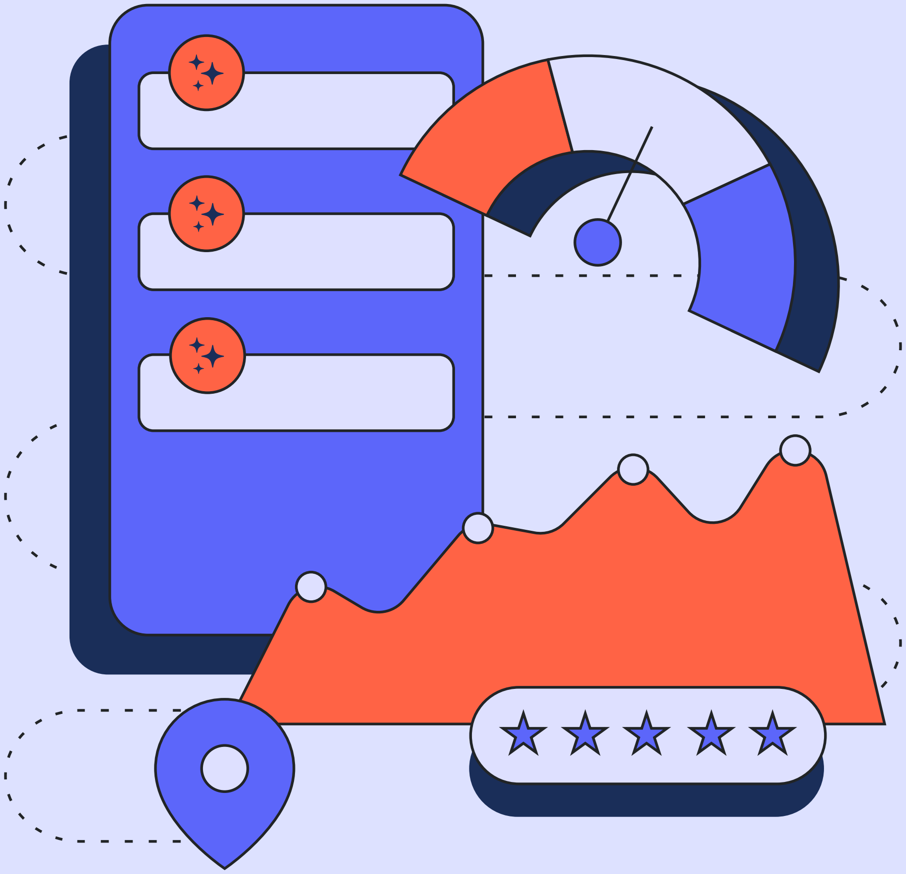
Software Engineering Manager

Engineering managers are keeping their team unblocked and streamlining processes wherever possible to decrease developer toil.

Generally, managers and their teams own the code and are responsible for actually approving bot-authored PRs. This can translate to hours of manual and repetitive work.

Measuring Your Bot-Generated PRs

LinearB's approach to tracking the volume and impact of bot-created code in your pipeline starts with PR labels. Every bot-created pull request is labeled, allowing metric tracking for this type of work. From there, you can compare process metrics against the unlabeled PRs.



"From 50% DORA health scores to 98% in less than 6 months. LinearB has given our business critical telemetry and attribution for our engineering teams. We've seen a 90% reduction in time to release resulting in a massive reduction in cycle time for our organization."

Matt Culver
Engineering Chief of Staff, Super.com

LinearB's SEI automation tool can auto-label PRs based on criteria such as:

1 The type of bot that created the PR (e.g. Dependabot)

2 The type of update (e.g. Major, Minor or Patch)

3 You can even label based on the libraries being updated in the PR.

→

Then

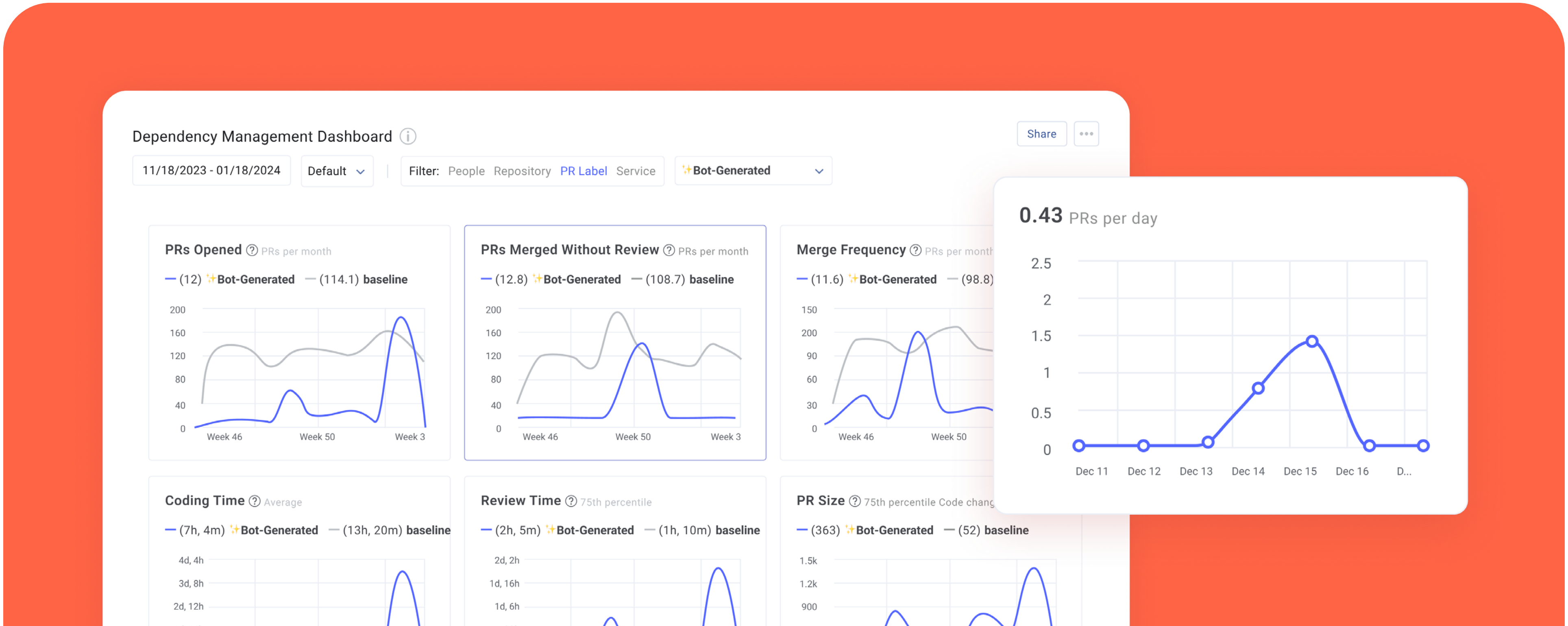
After selecting one of the above options for tracking bot-generated code, it's time to quantify the impact using engineering metrics.

Our Custom Bot PR Dashboard

Note that we've labeled all PRs with a Bot label in GitHub, and used this PR label in the dashboard filter definition. In this view, the blue line at the bottom represents all PRs with a Bot label, and the gray line represents all other PRs (our baseline in this case). We can now easily understand volume as well as key behaviors such as how long these PRs take to get merged.

Whenever we see a trend in our data that doesn't match our expectations, we drill deeper into our data by clicking on the spike in the graph. This will bring up a list of all the PRs that contributed to this datapoint, so we can take a closer look at all the bot-authored PRs slowing our team down.

Moving forward, we can also leverage SEI+ automation to auto-approve all safe bot-created PRs, removing toil and improving key metrics such as Cycle Time (more on this in the following section).



Dependency Management Dashboard

11/18/2023 - 01/18/2024 | Default | Filter: People Repository PR Label Service | Bot-Generated

PRs Opened — PRs per month
(12) Bot-Generated — (114.1) baseline

PRs Merged Without Review — PRs per month
(12.8) Bot-Generated — (108.7) baseline

Merge Frequency — PRs per month
(11.6) Bot-Generated — (98.8)

Coding Time — Average
(7h, 4m) Bot-Generated — (13h, 20m) baseline

Review Time — 75th percentile
(2h, 5m) Bot-Generated — (1h, 10m) baseline

PR Size — 75th percentile Code chang
(363) Bot-Generated — (52) baseline

0.43 PRs per day

Automating Your Bot-Generated PRs

Implementing SEI automation across your entire SDLC streamlines the release process and enables developers to focus on what they love—coding—instead of spending their time on manual merge tasks, data entry, and low-value code reviews.

The ability to merge bot-authored code faster and at a higher quality will enable your organization to gain a competitive advantage and deliver more value faster.

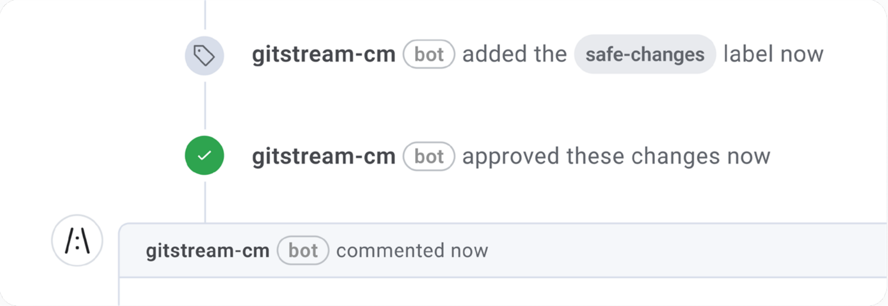
To this end, we recommend leveraging bot assistants that will enable your team to streamline their workflows in real time with automated support for critical tasks, goals, and projects.

With SEI automation, your team can do things like:

1. Ensure PRs are always linked to JIRA tickets
2. Auto-approve safe pull requests (such as updates to docs or image, etc.)
3. Get real-time PR updates with vital context (including estimated review time, PR owner, and related issue links)

Let's explore auto-approvals more thoroughly since this is where you'll be able to optimize your bot PR pipeline almost immediately.

LinearB's SEI automations can parse Dependency update PRs, to understand whether they are for Major, Minor or Patch updates. You can easily create a rule to auto-merge all patch update PRs in your repos, and another rule to auto-approve minor updates. If there are specific libraries that you want to exclude from this, that's an easy modification to the rules.



gitstream-cm bot added the safe-changes label now

gitstream-cm bot approved these changes now

gitstream-cm bot commented now

In the example, LinearB's SEI automation bot, gitStream, labels a minor update PR with the safe-change label and automatically approves it.

These automations replace rubber-stamping approval by devs with rule based rigor, saving toil, cutting down vulnerability times, and enabling measurement for dependency management workflows.

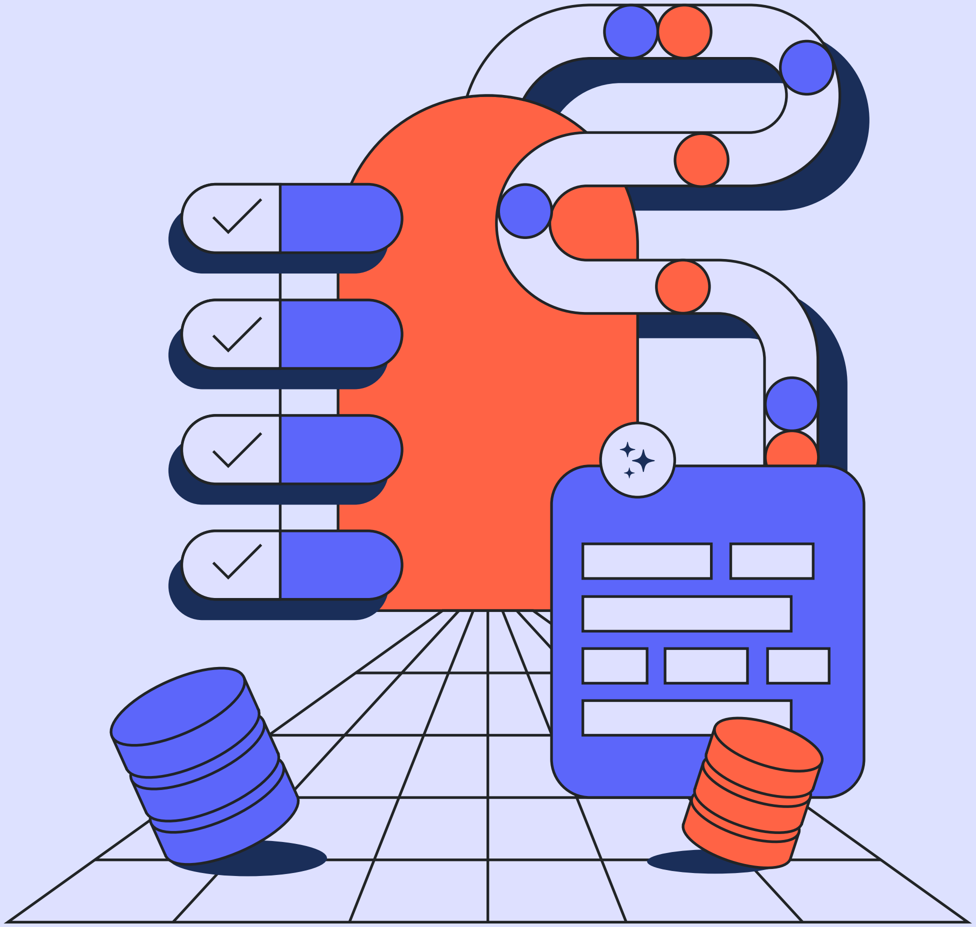
A study of key open-source repos shows a surge in bot-created PRs, with the share of bot PRs rising from **5% to 15%** over the past two years. This highlights a broader shift towards automation in software development.

The Future of AI Agents

Vendors like Dependabot and Renovate have pioneered this category. Their acquisitions in 2019 by Github and Mend, respectively, heralded the growing importance of automation in code management and compliance best practices.

A second wave of PR bots is coming, driven by advancements in AI agents and machine learning. The potential for automation to revolutionize code management becomes increasingly apparent.

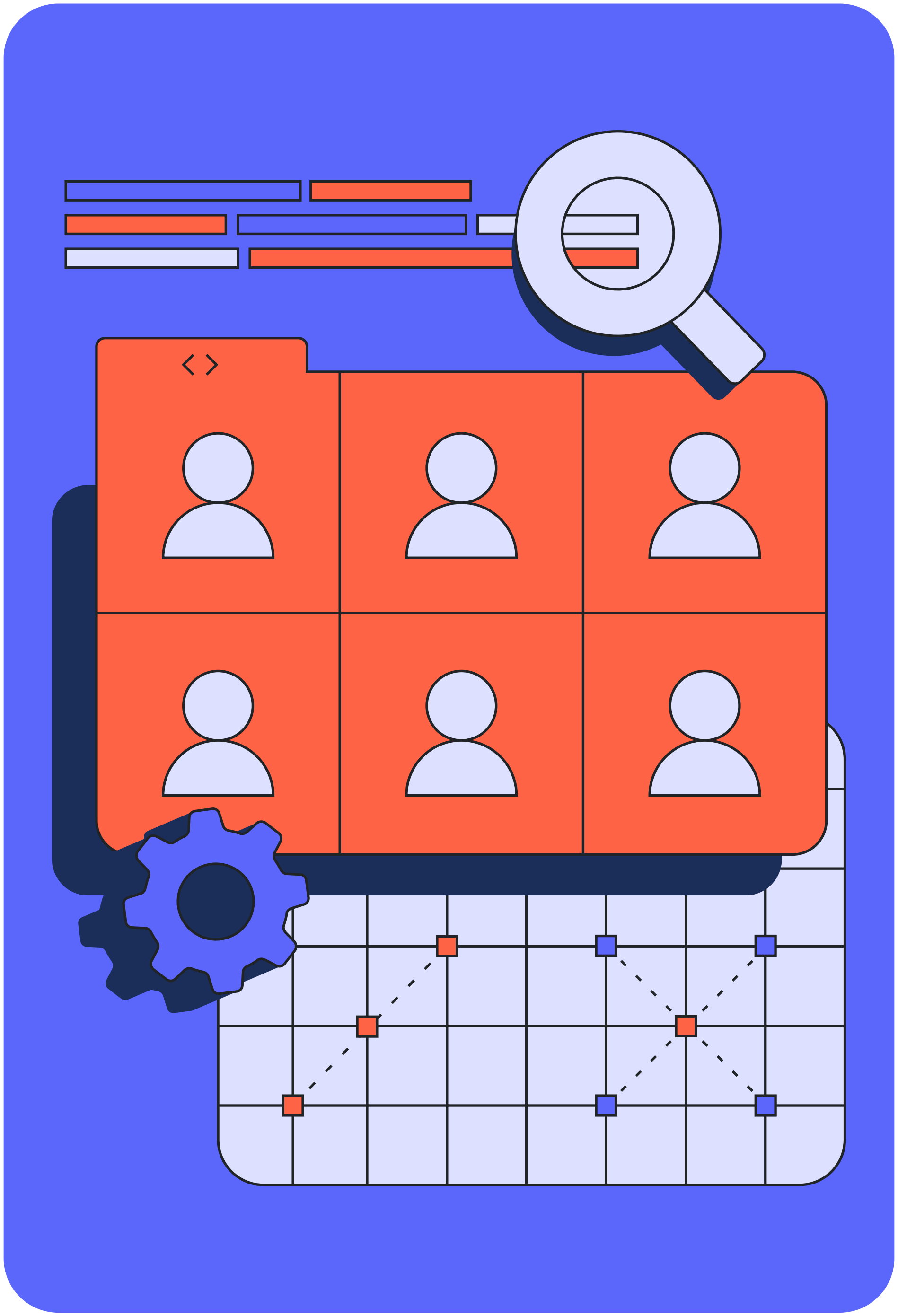
Sometimes referred to as Agentic AI, this future promises a new generation of even “smarter” bots that will be able to automate more complex tasks. Given the rise of AI-generated code, you can likely imagine a world in which the percentage of bot-authored code rises from 15% today to 50% or more in the very near future.



The potential for automation to revolutionize code management becomes increasingly apparent.

Now is the time to begin managing your bot-created PRs, while many of the changes are still straightforward. As bots become more advanced, they will offer more sophisticated solutions, ranging from solving security issues to improving code quality.

We're teetering on a future where AI-driven agents play a central role in the way engineering teams build and ship software. Embracing this change and preparing for the increased prevalence of bot- and AI-generated code will be vital for teams looking to remain competitive in a rapidly evolving industry.



"It's the companies that are best at software who are winning their sector. Being great at building software is directly relevant to how well you succeed in the market, and security is a big part of that."

Jeff Williams
CTO, Contrast Security

More Insights for R&D Leaders

See the science behind improvement.

2024 Software Engineering Benchmarks Report

The 2024 Software Engineering Benchmarks Report was created from a study of 3,600,000 PRs from 2,800 dev teams across 64 countries.

Download Report

State of DevOps Report

The DORA team at Google and LinearB have partnered to gather research from over 36000 professionals worldwide for the Accelerate State of DevOps Report.

Download Report

Engineering Leader's Guide to Developer Productivity

Discover how to quantify developer productivity, common blockers, strategies to improve it, and how and when to present dev productivity data.

Download Guide

↗

Learn more about Data-Driven Software Delivery