ActiveState

STATE OF CI/CD SURVEY 2020

TABLE OF CONTENTS

ACTIVESTATE CI/CD SURVEY

Continuous Integration and Continuous Delivery or Deployment (CI/CD) is an agile software development best practice designed to enable more frequent and reliable code updates. The CI/CD market is characterized by a large number of vendors with various on premise and cloud-based offerings, but no universal out-of-the-box approach exists.

Additionally, CI/CD remains aspirational for most organizations, who have a long way to go before they've fully automated the build, test, deliver and deploy cycle. ActiveState aims to help change that.

This survey was undertaken to help understand what works, what doesn't, and share best practices so we can all improve our CI/CD implementations. It will also help ActiveState identify the gaps our open source language automation platform can fill.

ACTIVESTATE AND CI/CD – WHAT'S THE CONNECTION?

With a 20+ year history of open source language support for organizations both large and small, we've seen some common pitfalls that may be negatively impacting CI/CD implementations, including:

Reproducibility
Ensuring that all systems are consistent throughout the CI/CD chain, as well as development and production.

Transparency
Understanding the original source for all artifacts throughout the chain can improve both security and compliance of production workloads.

Speed
"A hallmark of modern CI is spending 10 minutes to build a Docker image to run a process for 5s. [1]"

Here at ActiveState, we're researching how our open source language automation platform (the ActiveState Platform) can help improve CI/CD tooling and practices.

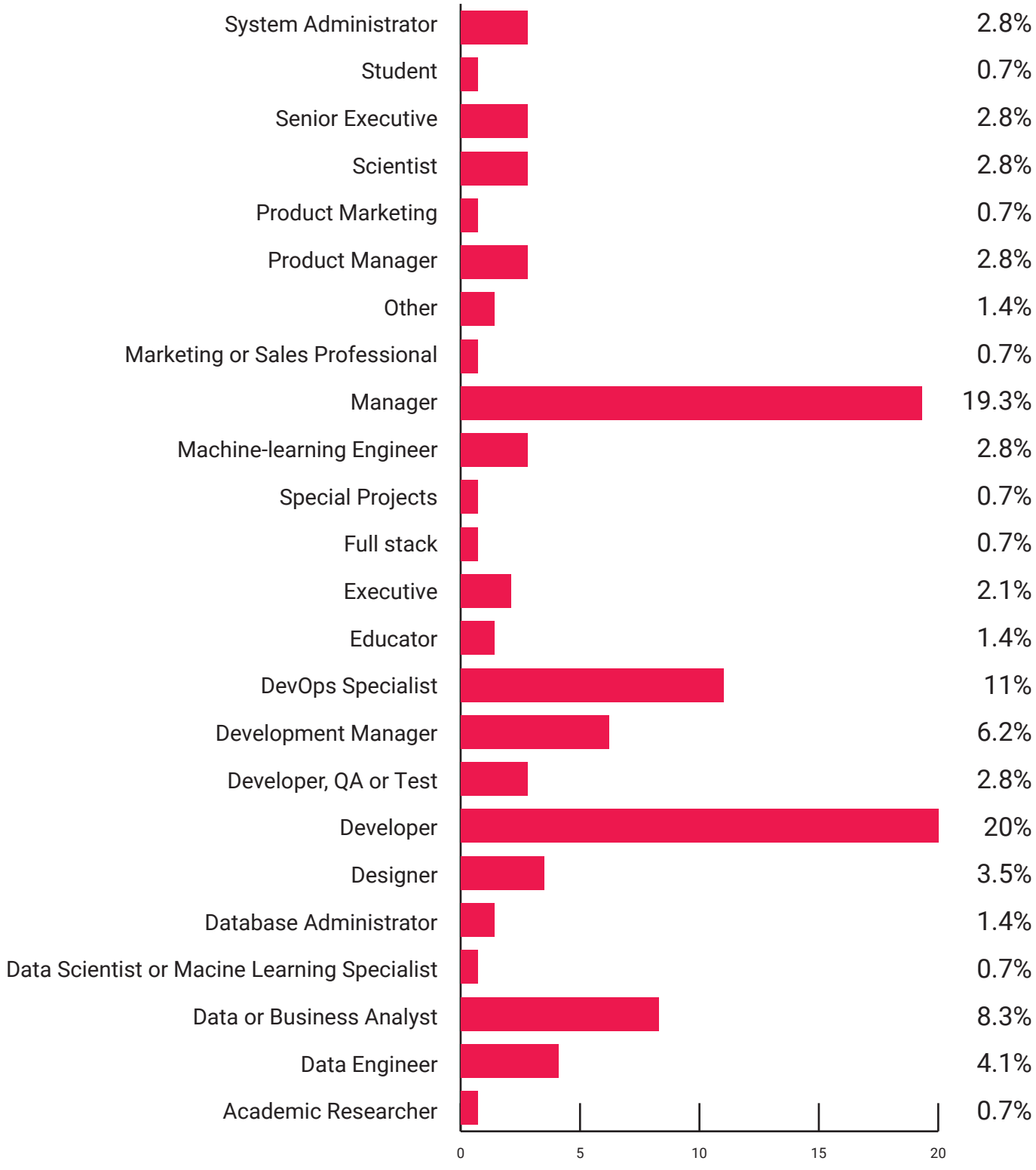
[1] https://twitter.com/indygreg/status/1231008674090344449

PART 1
DEMOGRAPHICS

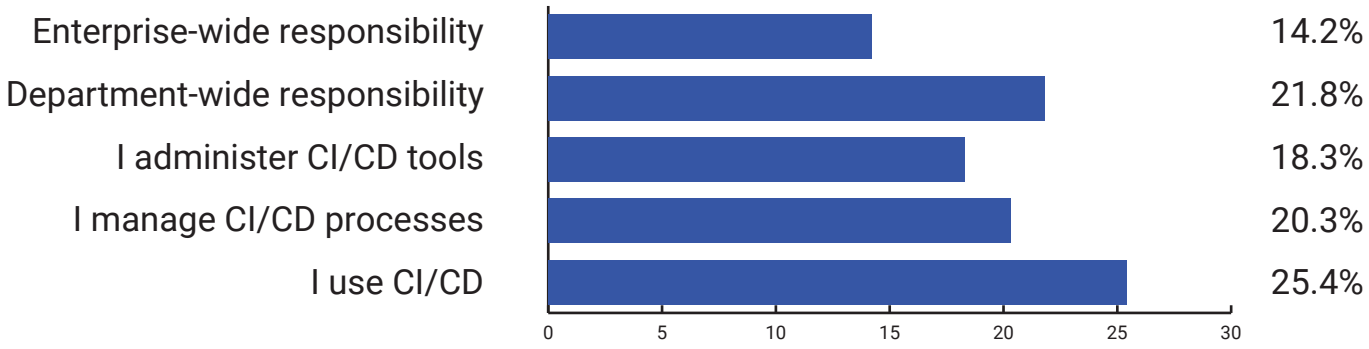
Respondents to ActiveState's CI/CD survey work in a wide range of industries at organizations of all sizes, and occupy a number of different roles associated with either running or managing CI/CD deployments.

While the majority of respondents have been working with CI/CD for at least a year, they have yet to fully implement their solution. Many organizations are wary about pushing directly to production every code change that successfully passes the CI/CD pipeline. Instead, most settle for continuous delivery to non-production systems where other events may take place, such as manual testing, beta testing, etc.

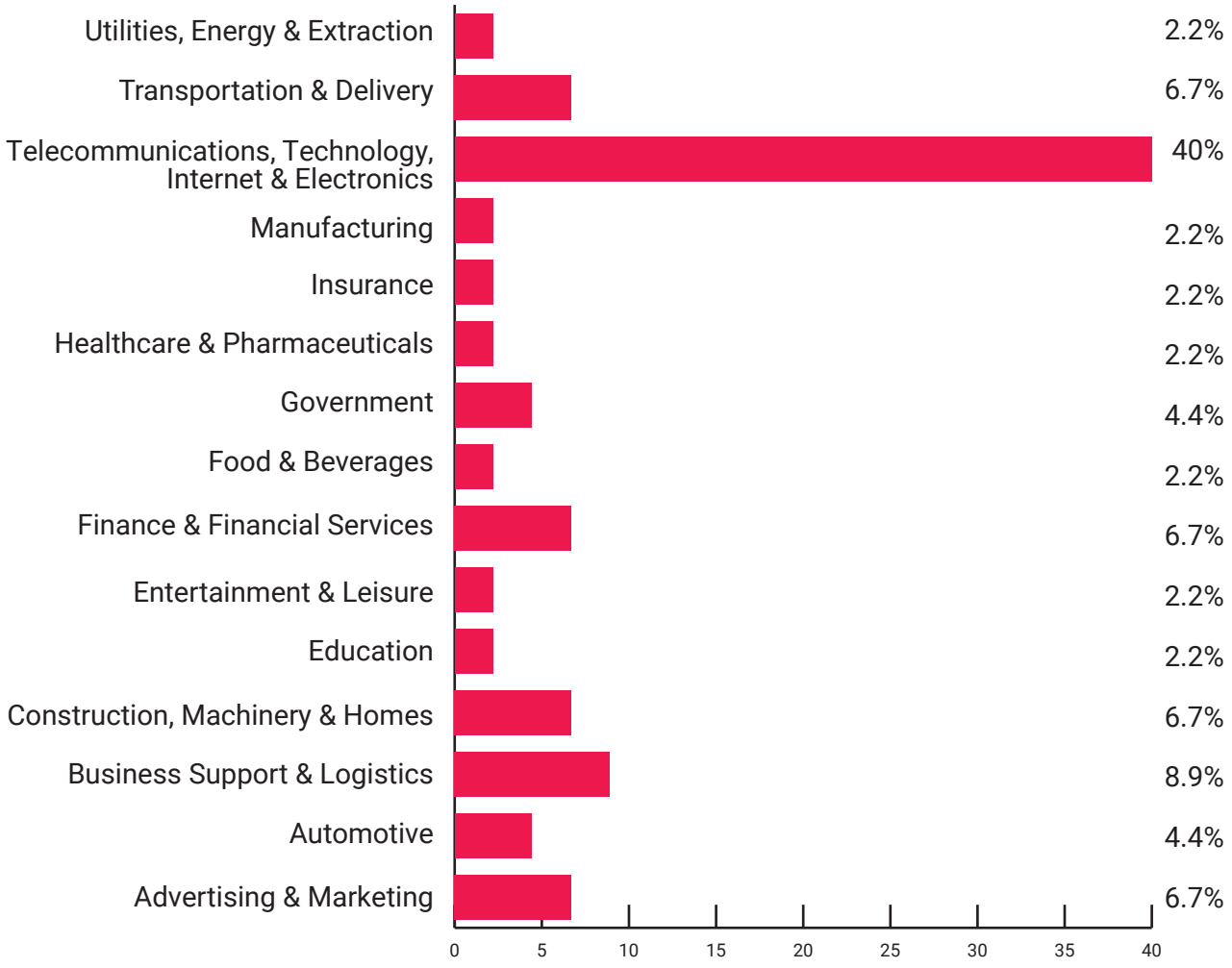
Which title best describes your role?



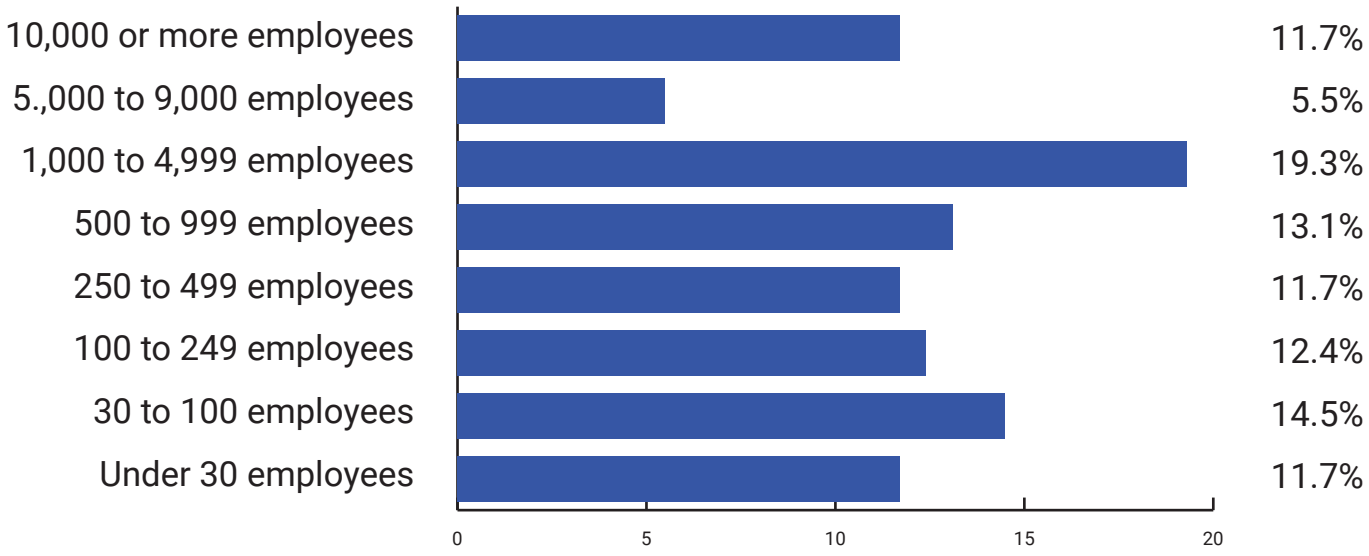
What best describes your responsibilities with respect to CI/CD?



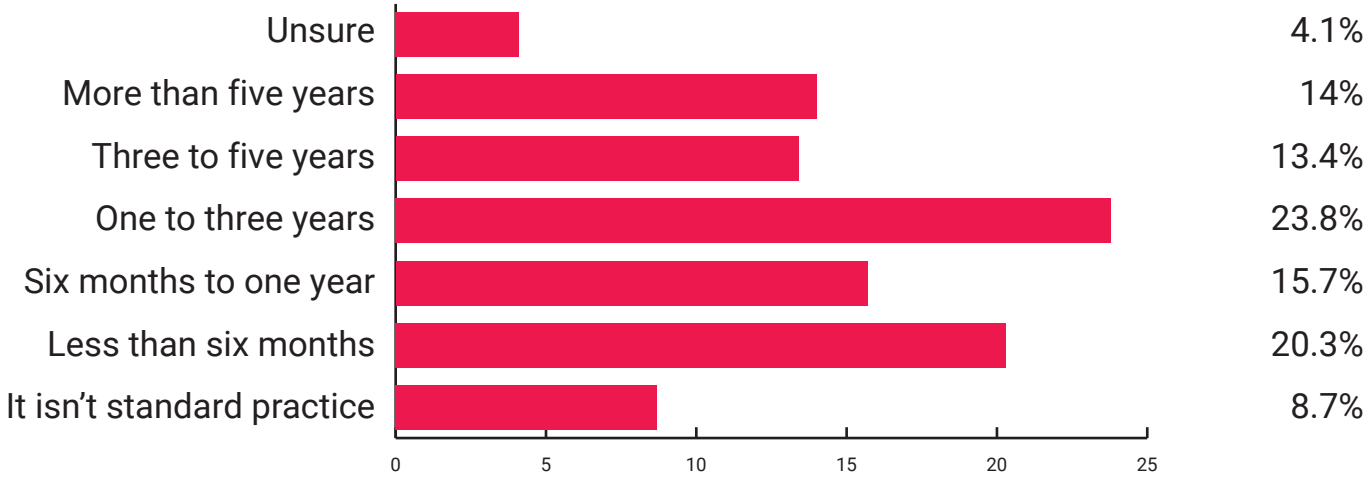
What is your organization's principal industry?



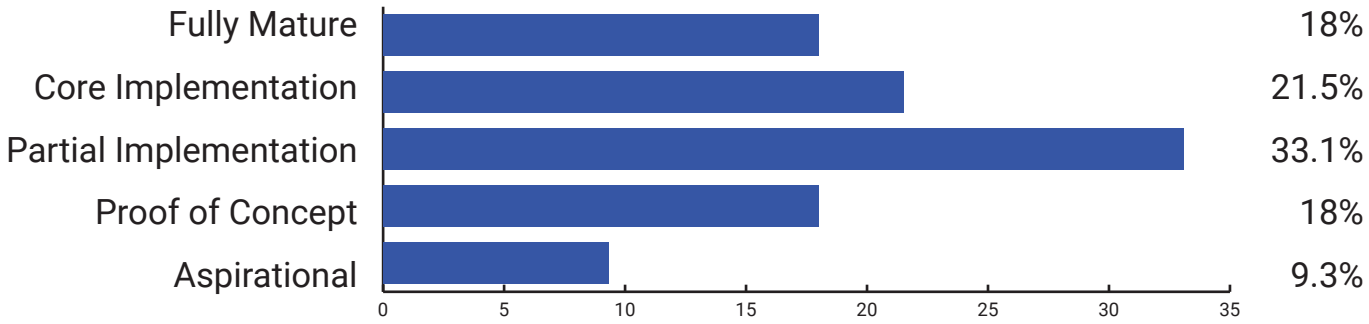
How large is your organization?



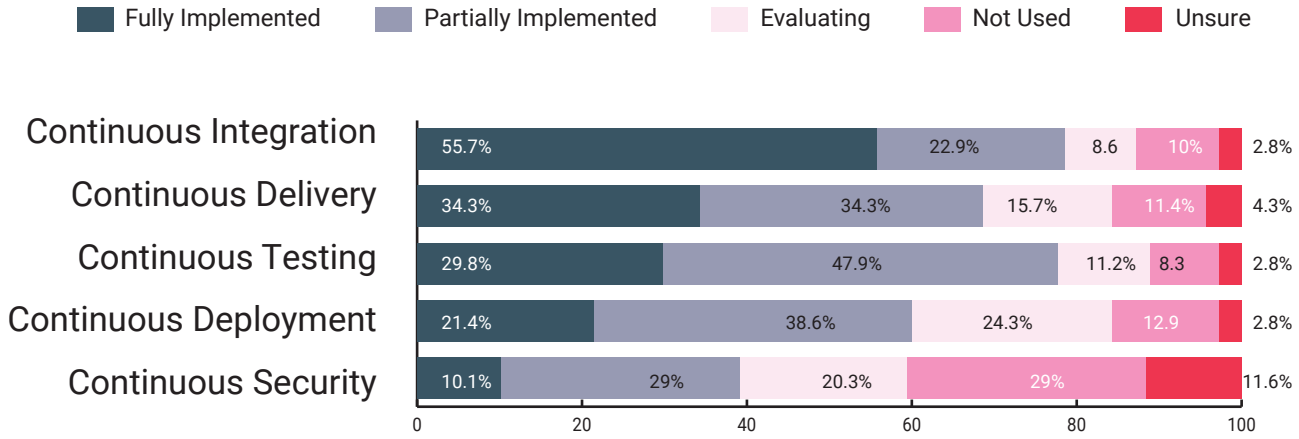
How long has CI/CD been a standard practice for your team(s)?



What best describes your team's CI/CD practice?



Which CI/CD best practices have you implemented?



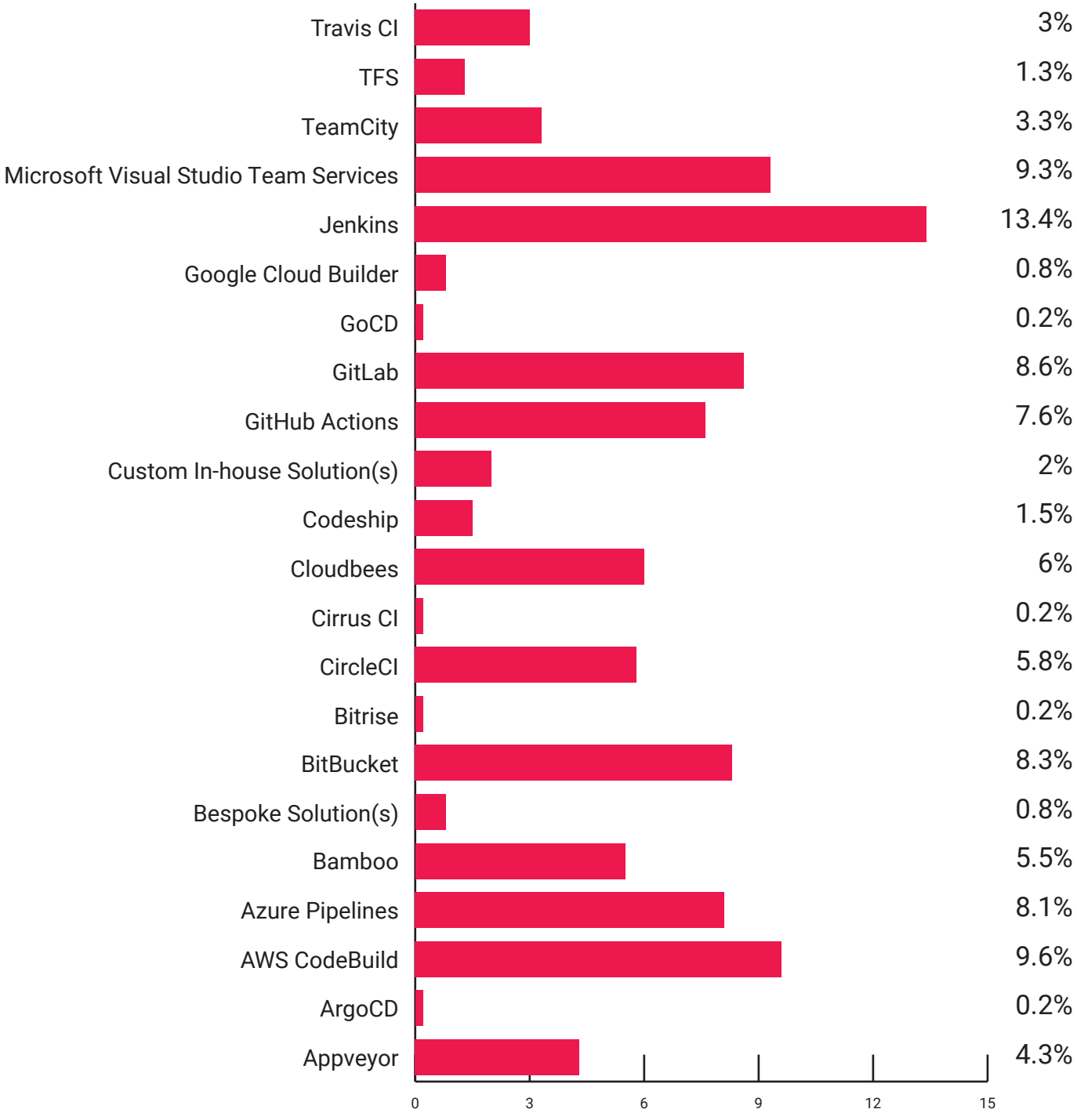
PART 2
TECHNOLOGY

Respondents use a wide range of both on premise and cloud-based CI/CD implementations. While there's a definite preference for Jenkins, respondents are open to trying different solutions, especially cloud-based ones.

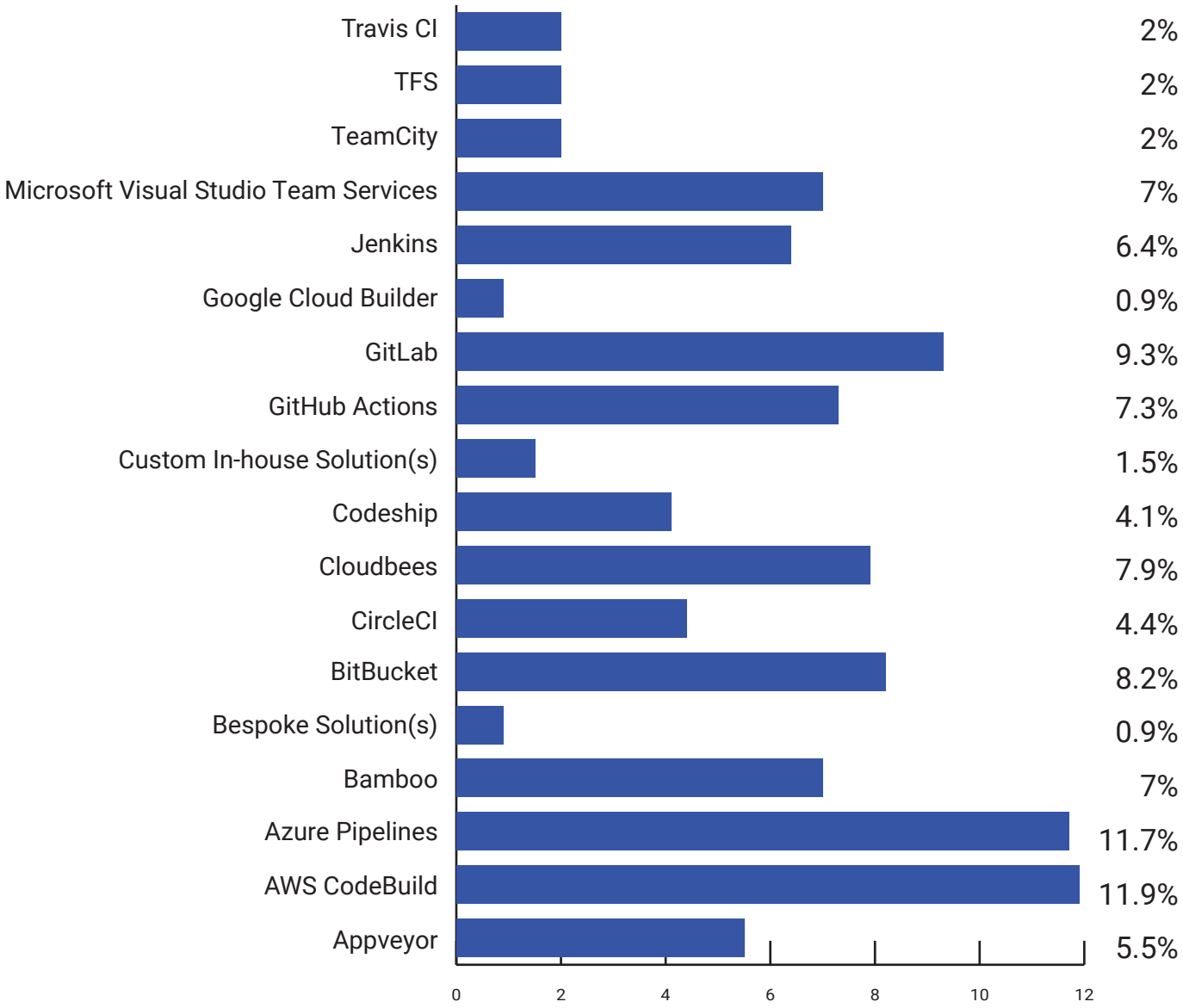
However the adoption of new CI/CD solutions is not straightforward. The majority of respondents estimated it would require between 4 months to 2 years to adopt a new CI/CD platform. One of the key reasons (besides license cost) is that, while many employ best practices such as the use of containers, there is no standard implementation methodology. Furthermore, using custom, non-standard runtime environments with cloud-based CI/CD solutions can be quite complex.

The ActiveState Platform provides pre-built, custom runtime environments that not only speed up container builds, but also simplify and standardize cloud-based CI/CD implementations, allowing you to create a far more portable solution. For more information, please refer to our "Optimizing CI/CD Implementations" white paper.

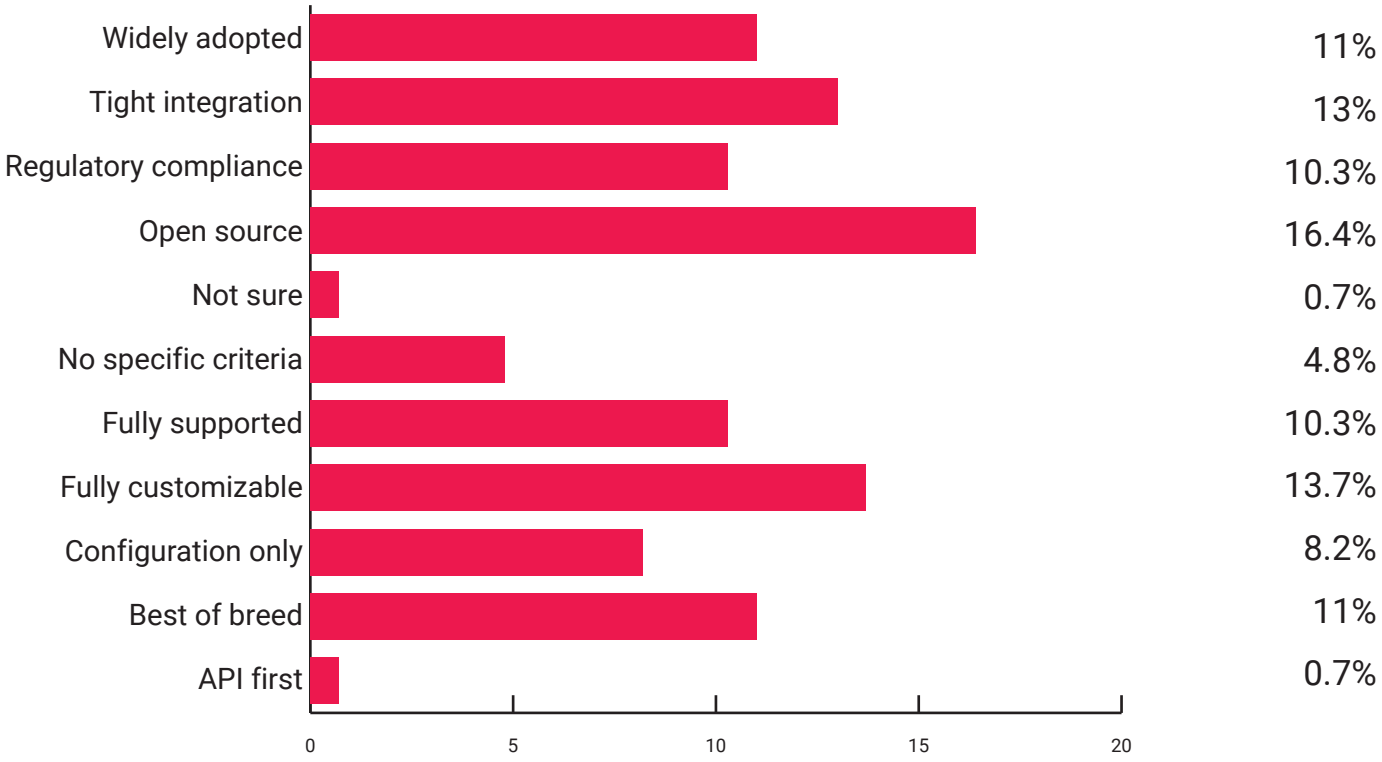
Which CI/CD tools do your teams currently use?



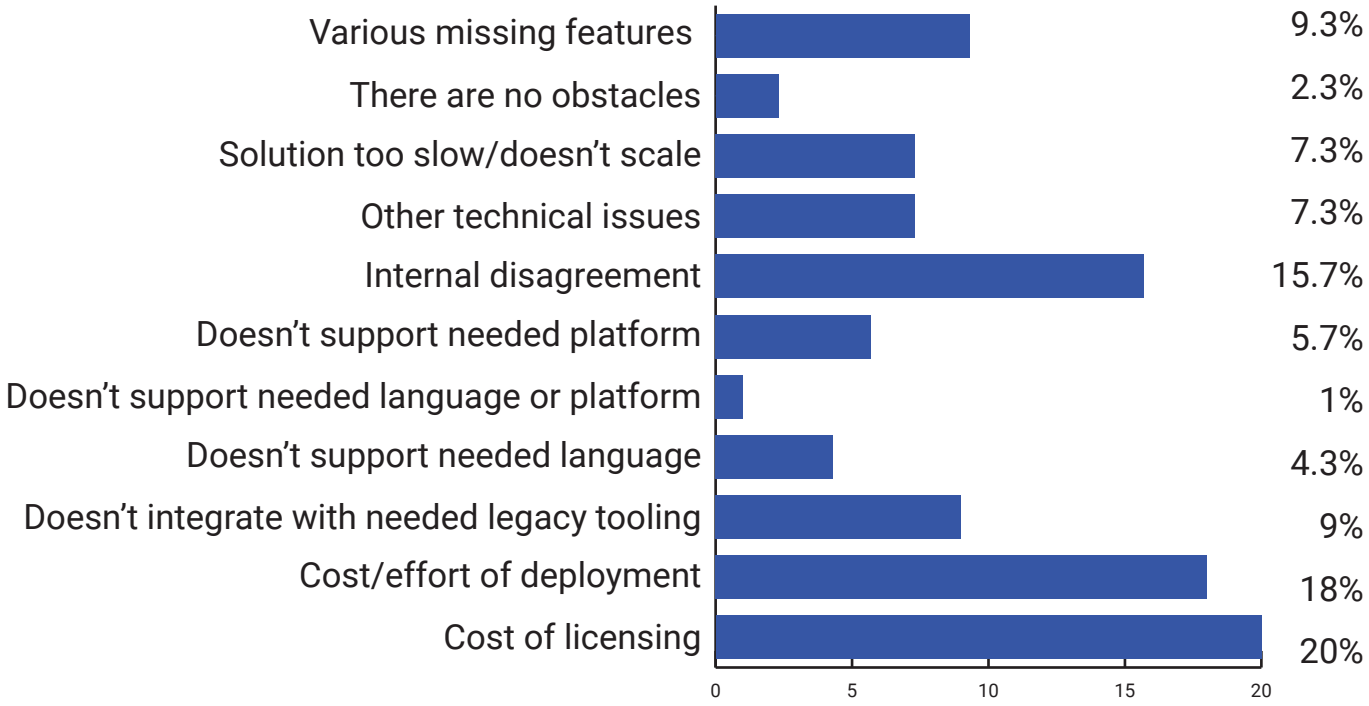
Which CI/CD tools do your teams want to adopt?



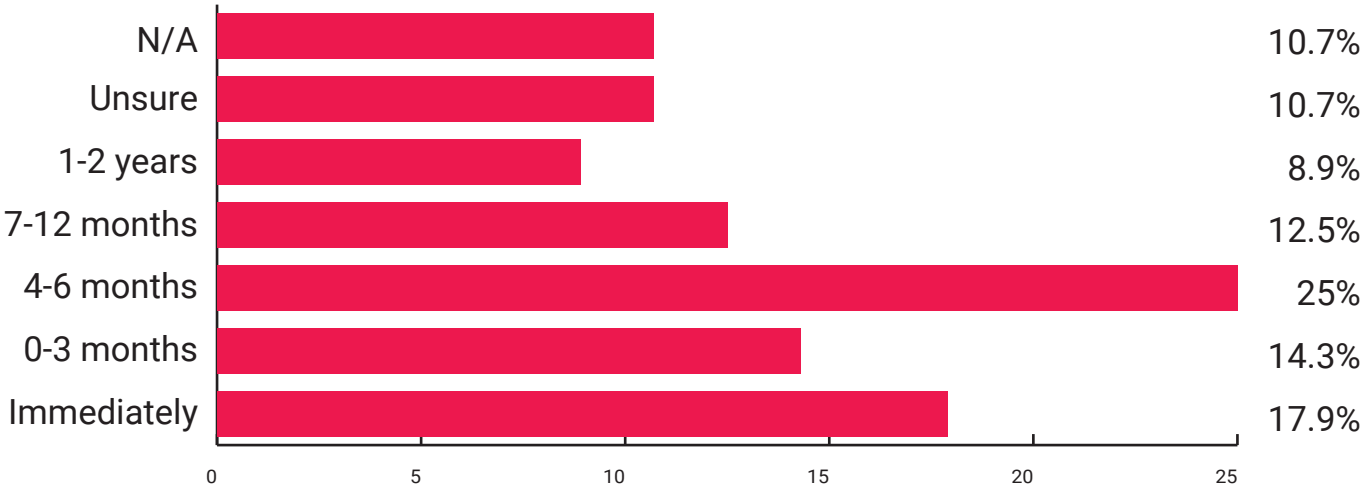
Which tool/vendor requirements are essential requirements?



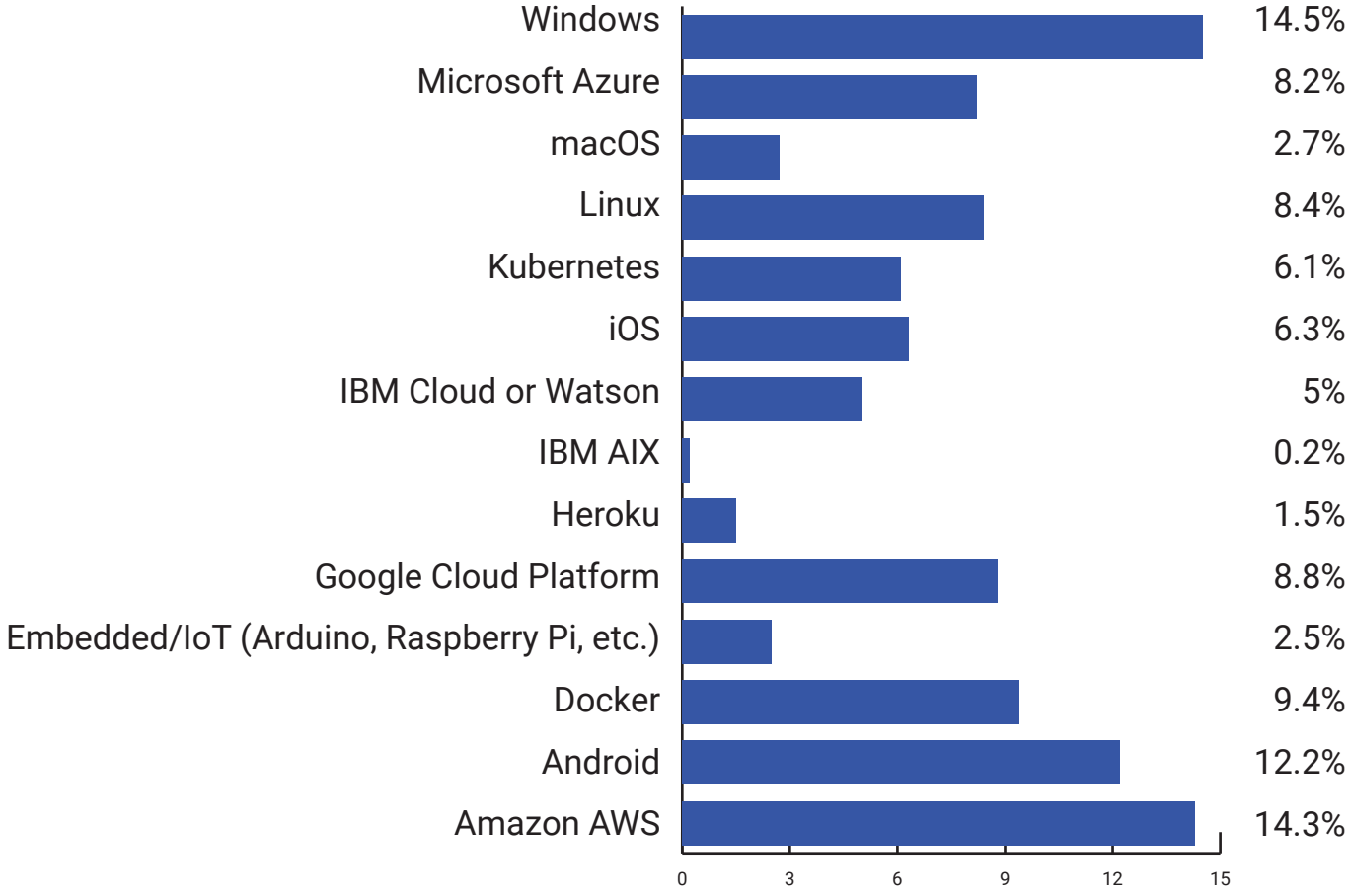
What is preventing you from adopting new CI/CD tools?



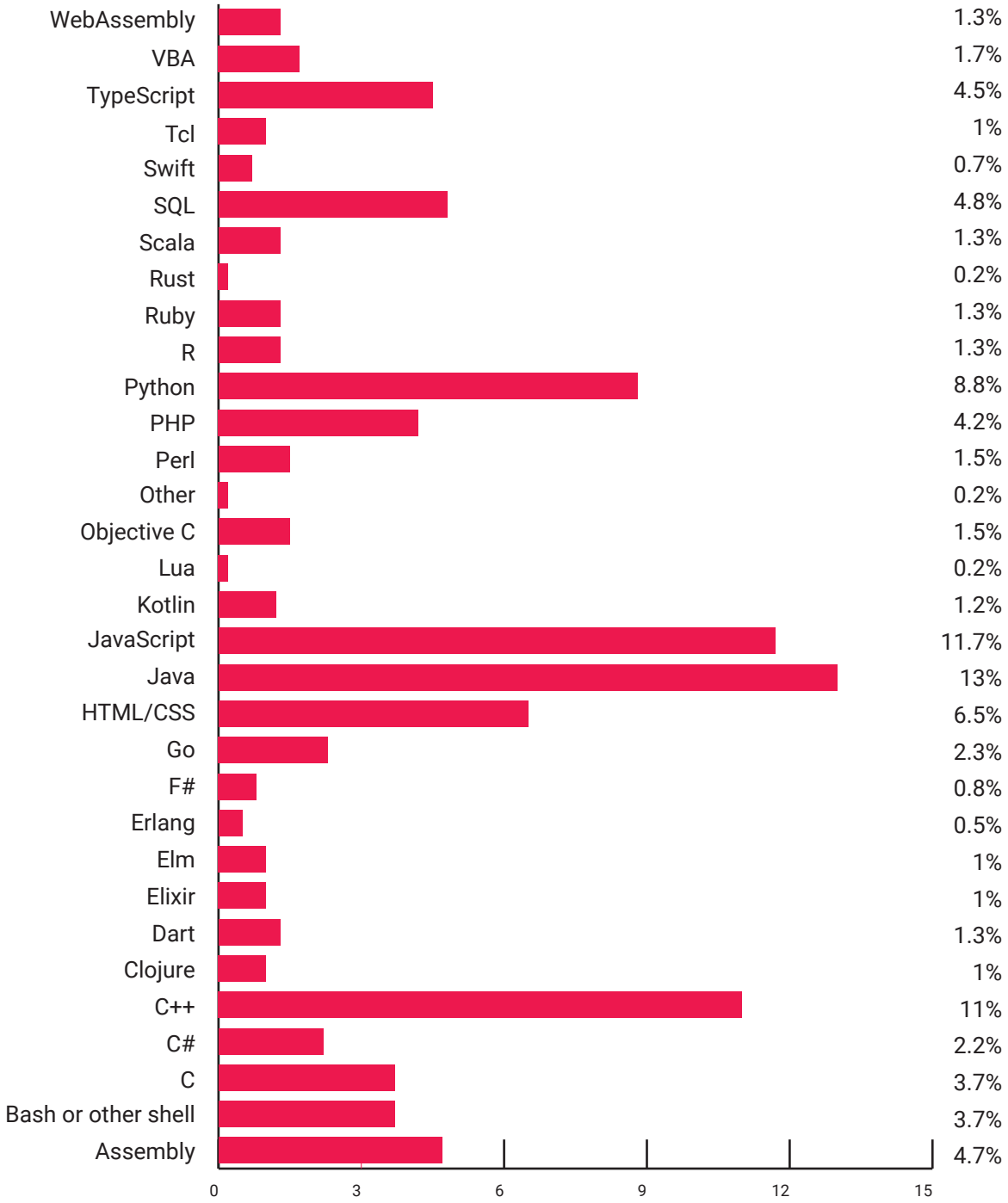
How long would it take to adopt new CI/CD tools in your organization?



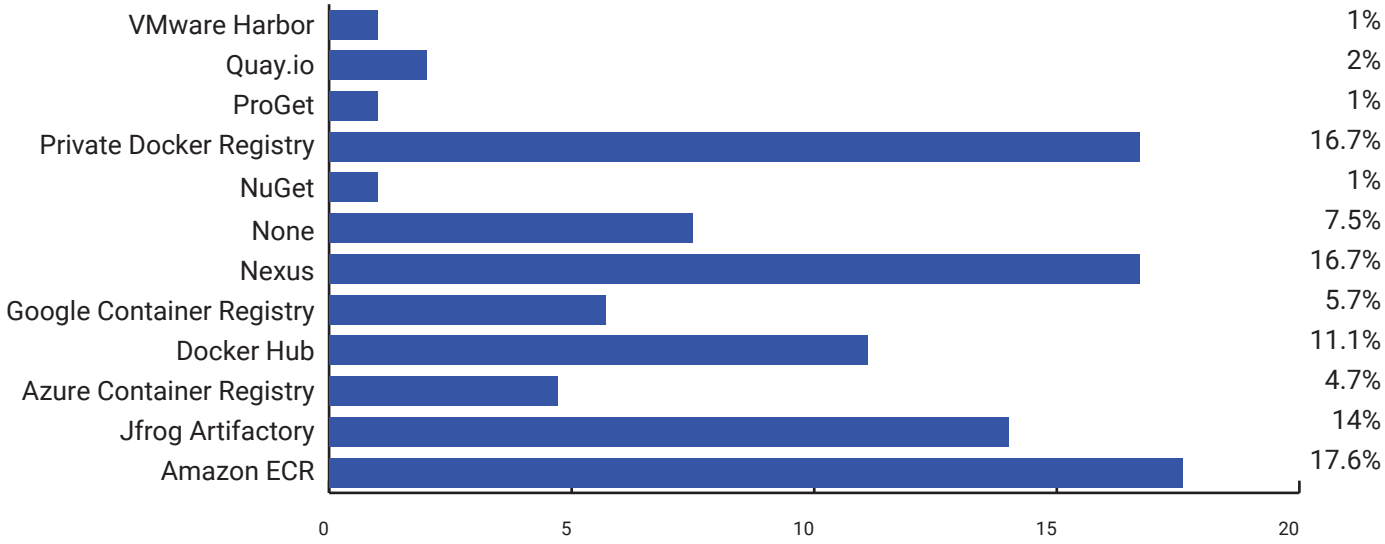
Which major deployment platforms does your organization use?



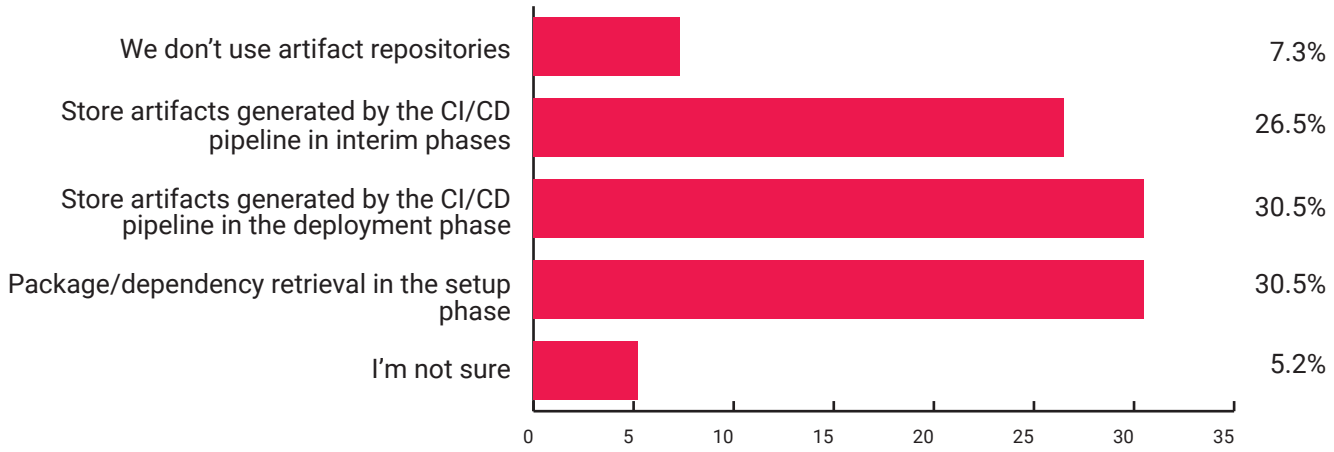
Which programming languages do you support in your CI/CD workflows?



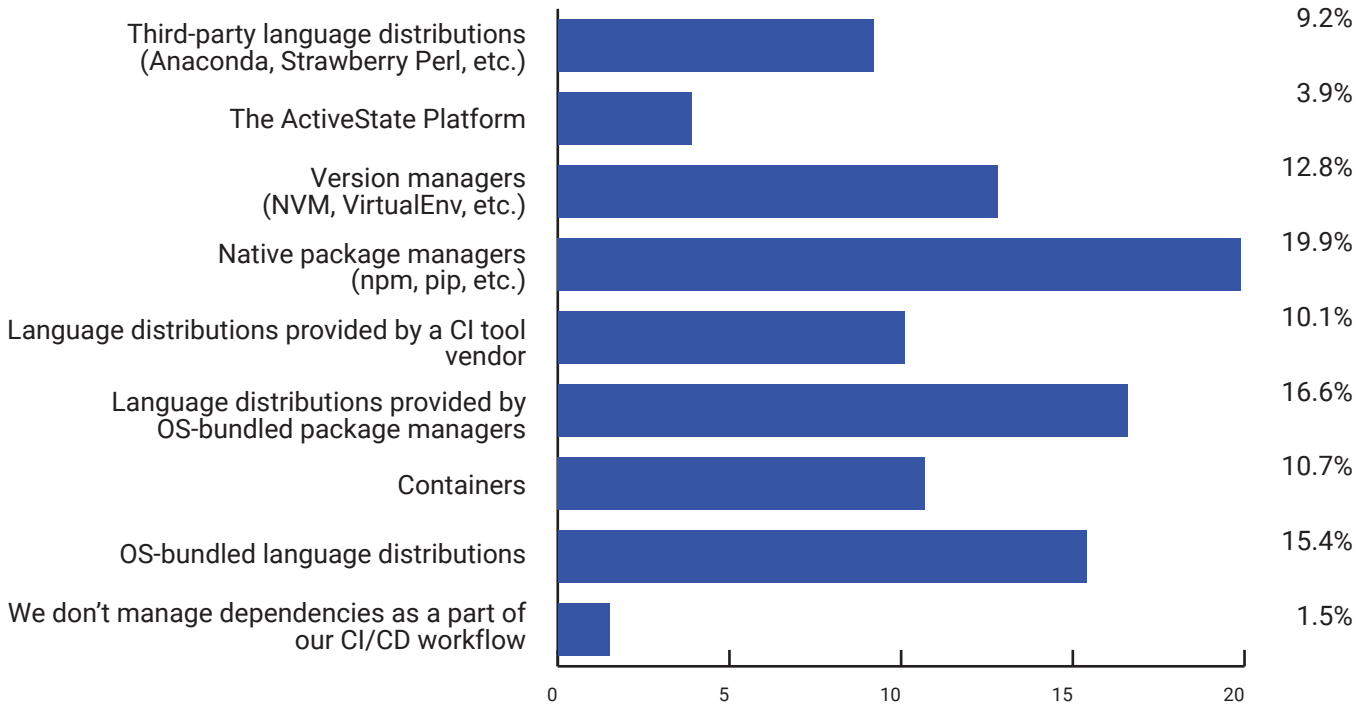
Which artifact repositories are used at your organization? Choose one or more.



How do you employ artifact repositories in a CI/CD context?



Which tools do you use to manage dependencies and create runtime environments in your CI/CD workflow?

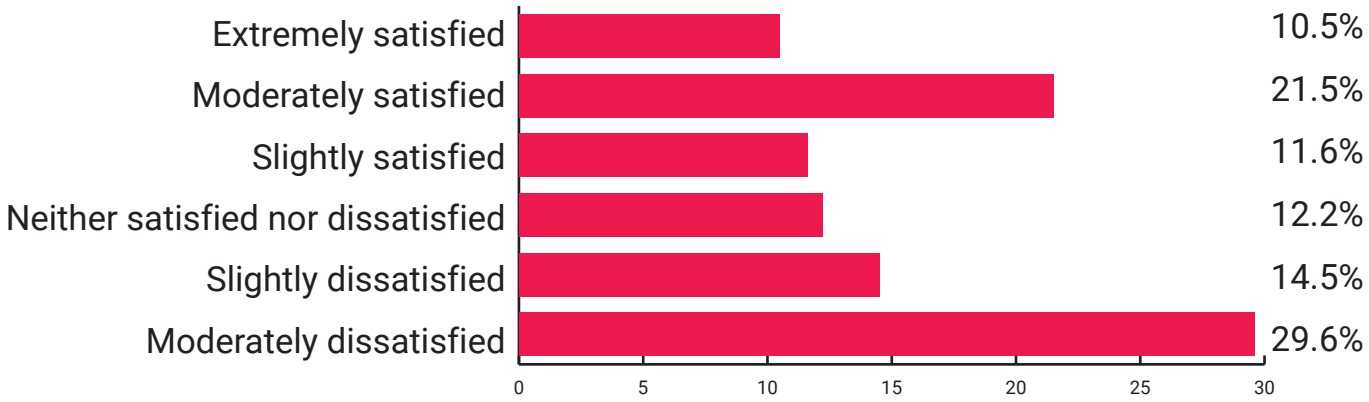


PART 3
KEY FINDINGS

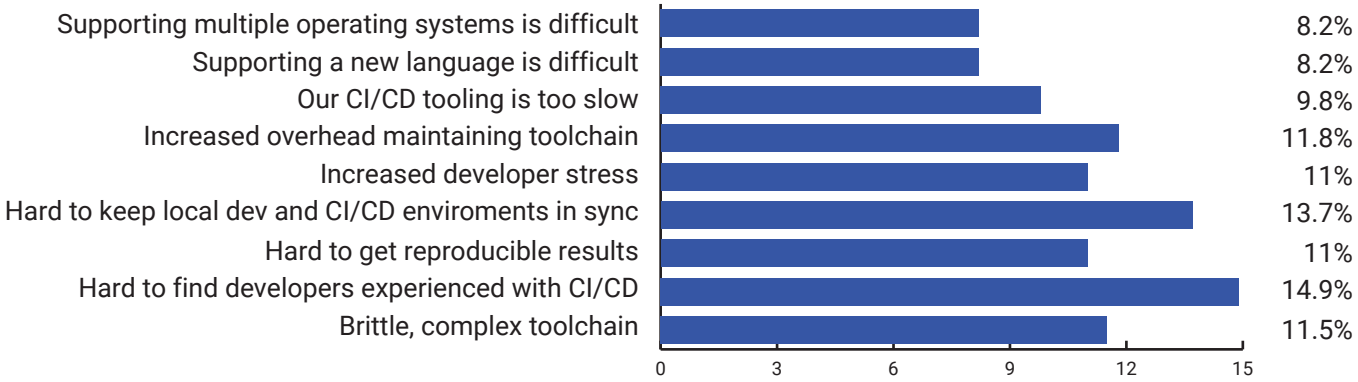
Only ~10% of our survey respondents would characterize themselves as being extremely satisfied with their current CI/CD implementation. Most respondents are still experiencing quite a few fundamental issues with their deployment, chief among them (besides finding experienced staff) are keeping development and CI/CD systems in sync, as well as working with multiple toolchains, both of which can lead to runtime inconsistencies.

While respondents widely use containers to help keep things consistent/in sync, if the container is not built in a consistent manner with the exact same manifest for both development and CI/CD systems, discrepancies can arise. This results in not only “works on my machine” errors, but also developers spending more time recreating the environment in which a bug was found than actually fixing the bug. For more information on these issues, please refer to our “Optimizing CI/CD Implementations” white paper.

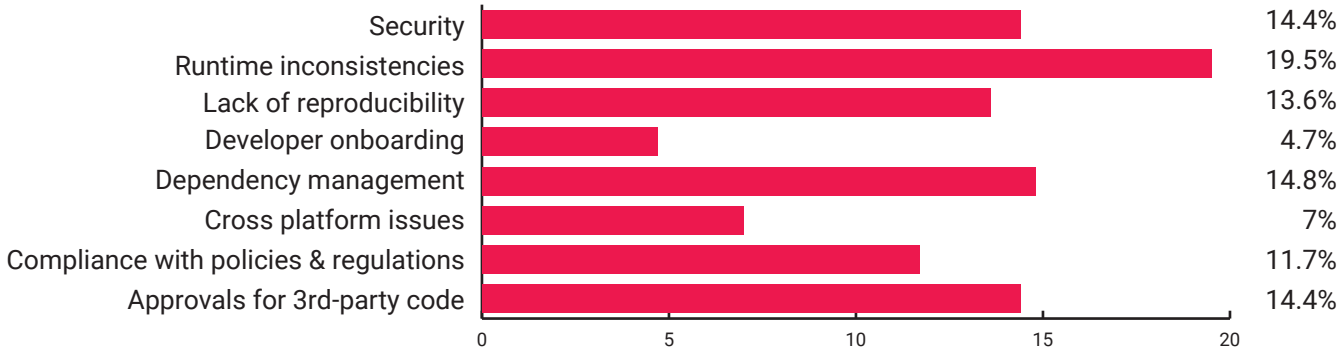
Overall, how satisfied are you with your CI/CD implementation?



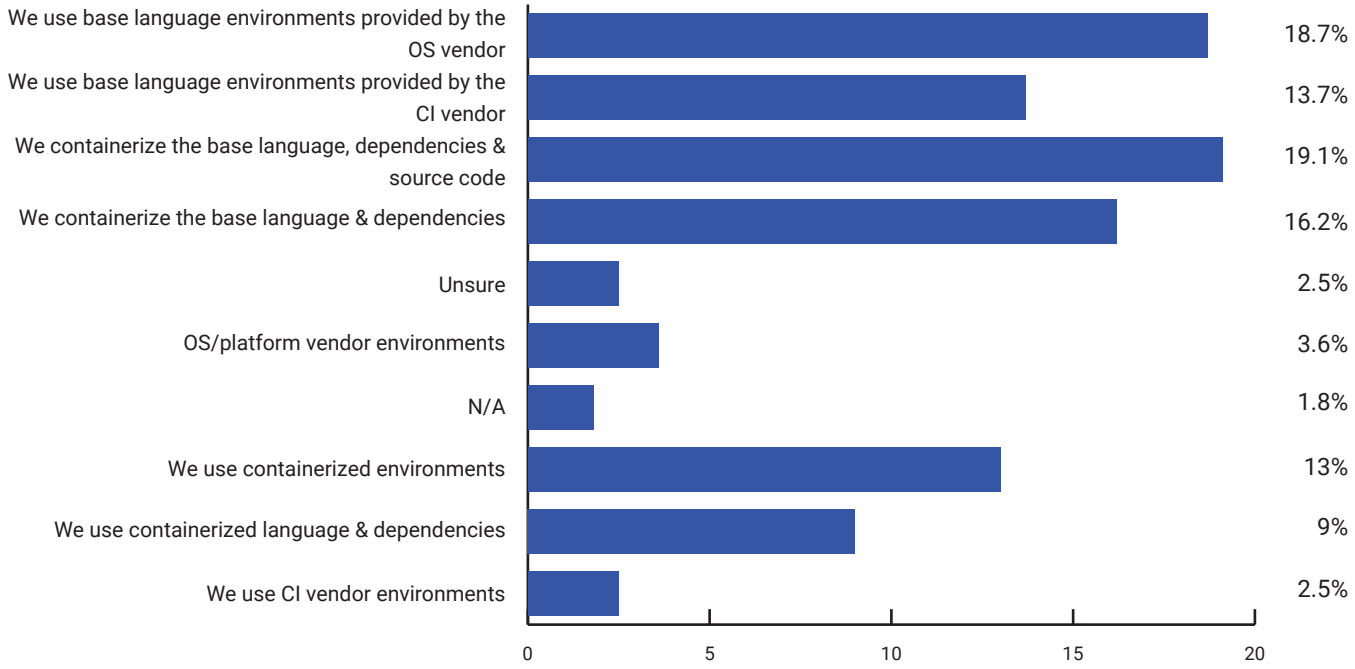
Which major drawbacks of CI/CD has your organization experienced?



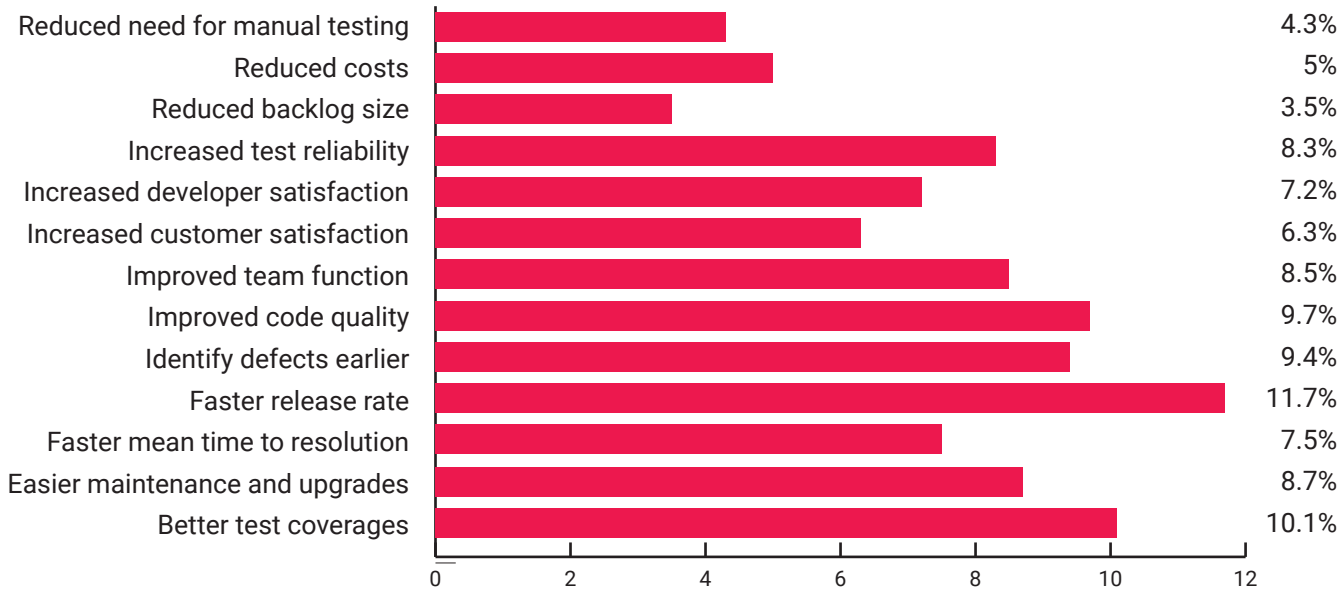
What are your top 3 challenges with managing language dependencies and runtimes?



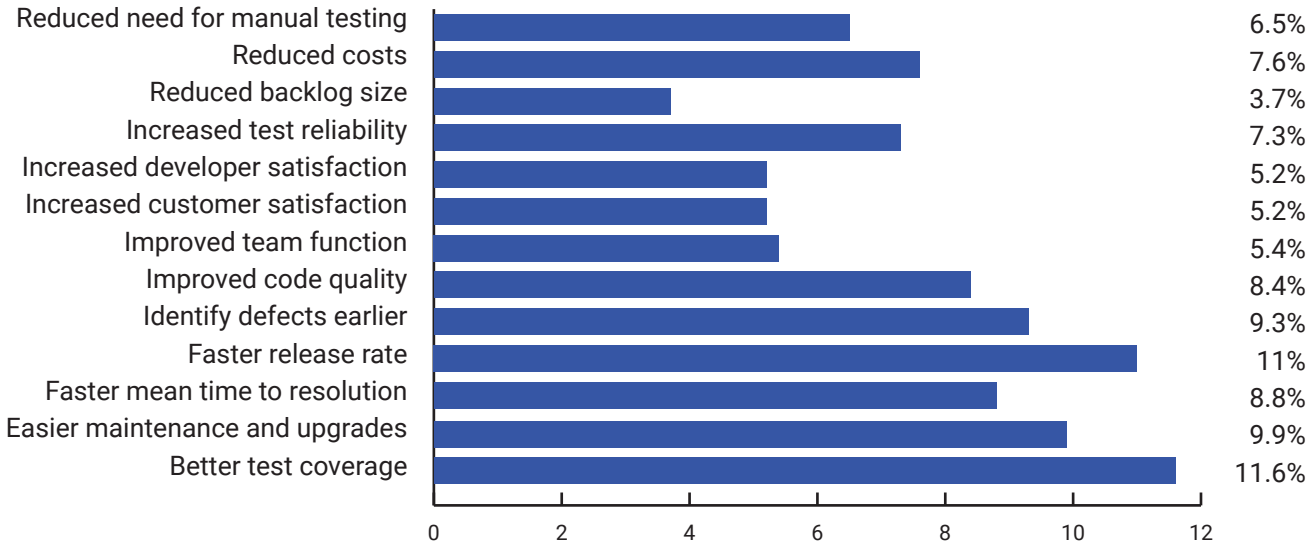
How do you currently manage language runtimes for your CI/CD workflow?



Which major benefits of CI/CD have you realized?



Which benefits of CI/CD did you expect but have not realized?



Conclusions

CI/CD is a key part of modern software development, yet no standard solution or definitive methodology exists. Additionally, most organizations have yet to fully implement their chosen solution, or fully realize the benefits from it. Some of the major problems organizations face with their CI/CD implementation include:

- Hiring experienced CI/CD developers
- Inconsistencies between development and CI/CD systems
- Dependency management
- Synchronizing development and CI/CD environments
- Toolchain complexity and overhead

The ActiveState Platform can help organizations with many of these issues by providing:

A cloud-based, unified toolchain for Windows, Linux and macOS that supports Python, Perl and Tcl implementations.

A single, central source of truth for all runtime environments, eliminating discrepancies between development and CI/CD systems, and ensuring they can remain in sync.

The ability to pull a pre-built runtime into CI/CD pipelines, simplifying dependency management, promoting consistency and speeding build times.

To learn more about the ActiveState Platform, sign up for a free account and try it yourself, or read more about ActiveState's CI/CD capabilities.

ABOUT **ACTIVESTATE**

ActiveState helps enterprises manage the complexity and risk of using open source languages at scale, while giving developers the kinds of tools they love to use. More than 2 million developers and 97% of Fortune 1000 enterprises use ActiveState to support mission-critical systems and speed up software development while enhancing oversight and increasing quality. For more information, please visit www.activestate.com.

info@activestate.com
www.activestate.com
@ActiveState
/activestate
/activestatesoftware