VERACODE EBOOK

5

FIVE PRINCIPLES FOR

Securing DevOps

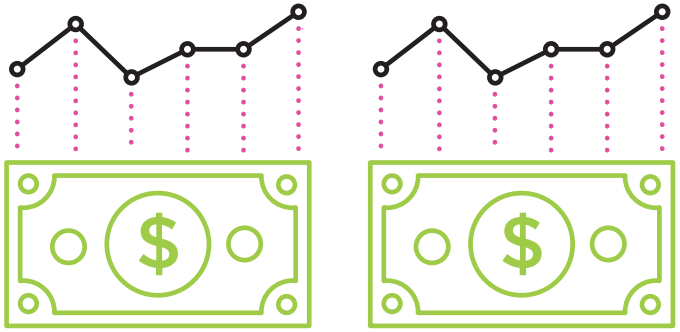
VERACODE

INTRODUCTION

DevOps, a new organizational and cultural way of organizing development and IT operations work, and its sister technologies, continuous integration and continuous deployment (CI/CD), have transformed the way we create software.

And there is widespread evidence that DevOps practices, despite their substantial organizational, cultural and technological requirements, are spreading rapidly.

In fact, there are sound business reasons for executives to embrace these changes.



A recent study shows that firms with high-performing IT organizations are twice as likely to exceed their profitability, market share and productivity goals.

Forsgren, N., J. Humble (2016). "DevOps: Profiles in ITSM Performance and Contributing Factors." In the Proceedings of the Western Decision Sciences Institute (WDSI) 2016, Las Vegas, NV. Available at SSRN: ssrn.com/abstract=2681906

Further, specific disciplines of continuous delivery, including test and deployment automation, trunk-based development, continuous integration and version control of app and system configuration, all lead directly to higher levels of IT performance and, therefore, to higher levels of organization performance.

But reaping these gains requires rethinking application security. To secure DevOps, it is critical to understand how DevOps and CI/CD are different from Agile development and how this difference changes the requirements for application security solutions. It is also important to recognize that, as CI/CD in particular continue to evolve, so do the requirements for application security.

THIS PAPER

- ✔ Provides background on the evolution of DevOps.
- ✔ Proposes five principles that solutions seeking to integrate application security into DevOps and CI/CD must address.

DevOps

Evolution and Revolution

DevOps seeks to enable software development teams to more consistently hit or exceed their goals for on-time delivery of high-quality software that meets the needs of the business. It does this by removing organizational barriers between Agile development teams and non-Agile supporting processes.

Many Agile software projects have succeeded in improving their quality practices only to face the reality of failed deployments when unanticipated operational requirements resulted in software that did not meet the needs of availability, scalability or manageability. By integrating activities and organizations like operations earlier into the development process, DevOps seeks to expose the development team to these potentially surprising or disruptive requirements early so that the team can plan for and address them ahead of time.

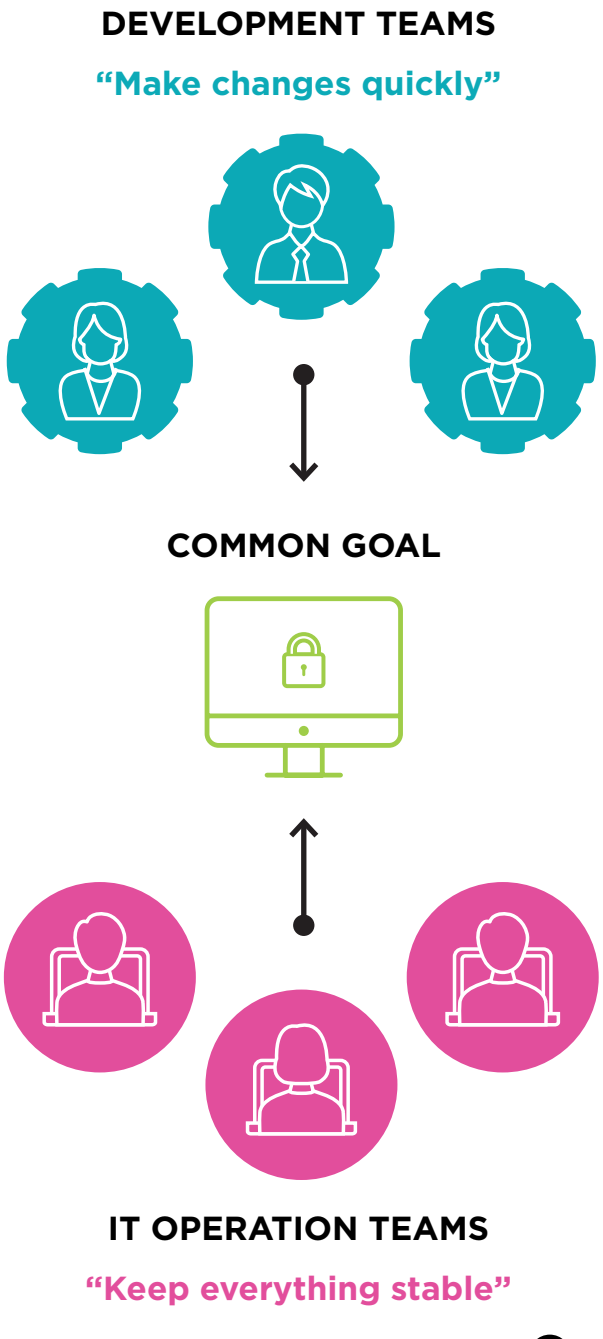
The process of bringing other teams, in particular operations, into the development process began as a revolt against heavyweight and highly manual operations practices that were seen as slowing development down.

DevOps thought leader Gene Kim has stated that DevOps practices explicitly seek to align the potentially at-odds goals of "make changes quickly" (development) and "keep everything stable" (IT operations) by bringing the teams together and giving them shared responsibility for software delivery and operation. This organizational alignment supports all the other activities of DevOps.

In this way, DevOps is a natural evolution of Agile software development and its culture of "retrospectives," "do better" and clearing blockages to getting work done. But the specific manifestations of this cultural and organizational change have been revolutionary for how software is built, beginning with how — and how frequently — it is delivered to market.

FOR INSTANCE, DEVOPS:

1. Embraces an existing software development trend, continuous integration, and its transformation into continuous deployment.
2. Implements insights from traditional manufacturing quality control processes to the software development process.



If DevOps includes cultural, organizational and technological components, continuous integration and continuous delivery, or CI/CD, is the technological foundation on which DevOps builds its practices.

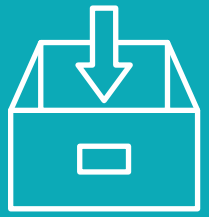
CI/CD seeks to automate much of the routine work of transforming code changes into working software, including delivering tested code into production. From its roots in build servers like Hudson, Jenkins and Microsoft Team Foundation Server, CI/CD has become a collection of technologies and practices that supports the integrated mission of releasing new code changes while keeping things stable.

Technologies that allow DevOps organizations to move faster include:

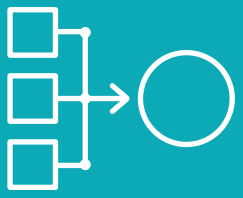
AUTOMATED BUILD AND VERIFICATION OF CODE CHANGES

UNIT TESTS

MICROSERVICES



CONTAINERIZATION



TRUNK-BASED DEVELOPMENT FEATURE TOGGLES

OPERATIONAL MONITORING

“Shifting Security Left” Drives New Requirements for AppSec

Like operations, security’s goals of minimizing enterprise risk sometimes seem to be at odds with development’s mandate for change. In reality, there is a middle path that can allow development to deliver more secure code at DevOps speed, but it requires security to adapt to the principles that have proven successful for DevOps.

Considering the goals of CI/CD helps us identify the following five principles for securing DevOps:

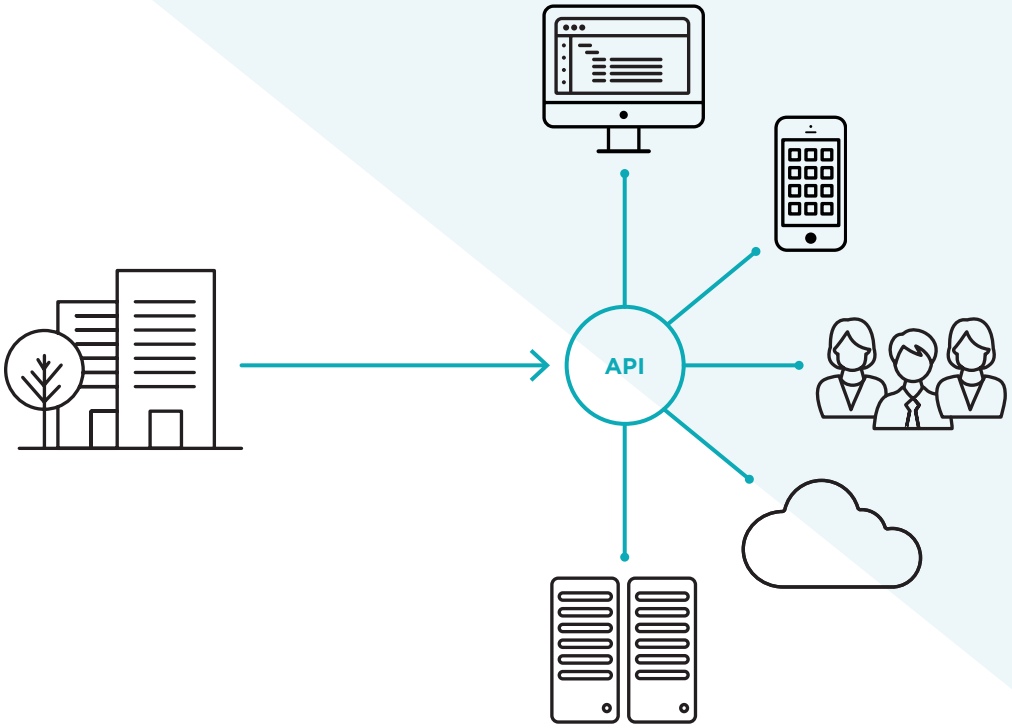
1.  Automate Security In
2.  Integrate to “Fail Quickly”
3.  No False Alarms
4.  Build Security Champions
5.  Keep Operational Visibility

PRINCIPLE ONE

Automate Security In

1

Automated invocation of security testing requires a comprehensive API to initiate, control and return results from software testing, and should include productized support for common tools of development teams.



PRINCIPLE TWO

Integrate to “Fail Quickly”

2

Integrating security into the CI/CD pipeline ensures that security testing happens with every release, and avoids the problem of leaving application security entirely in the hands of the developer or as a step late in the process.

There are several ways to address this requirement, for instance:

- **Scan small units of code so that results can be returned within the latency tolerance of the existing process in the pipeline.**
- **Allow the pipeline to kick off tests and feed the results into the backlog of the development team outside of the pipeline, essentially conducting the full application test in parallel.**

Regardless of how you integrate static testing into the pipeline, full application testing is still necessary: security issues may be introduced into the code that can only be found via a full program analysis. You can conduct full application tests outside the scope of the pipeline, or only on builds that make it to a certain stage of release candidate qualification.

In addition, you don’t need to stop at integrating with the pipeline. The best way to catch software defects quickly is to introduce tests that run as close to the developer as possible — for example, with quick-running tests triggered on check-in or even as pre-check-in gates. You can also allow developers to quickly test from the IDE.

PRINCIPLE THREE

No False Alarms

As the industry has learned, a technology that reports too many false positives will be ignored and will fail to be adopted. This is doubly true in CI/CD, where a failed security test may stop a critical business function from being delivered to production — or a critical patch from being released. That may be tolerable if the security issue is real, but is completely intolerable if the finding is a false positive.

3

PRINCIPLE FOUR

Build Security Champions

Most developers are not trained in the practices of secure coding. But doing so gives the security team a force multiplier and reduces culture conflict by embedding application security knowledge directly in the team.

4

PRINCIPLE FIVE

Keep Operational Visibility

5

Application security cannot stop after deployment. As with other aspects of DevOps, a well-engineered solution must support "closed loop" feedback from production in the event of a security incident. There are several scenarios in which operational visibility into application security is particularly important.

1

TO ENABLE THE TEAM TO DEPLOY FASTER.

The business may choose to trade full application security testing for faster deployment and, therefore, rely on the ability to test after deployment and quickly update if an issue is found.

2

TO CATCH EXCEPTIONS.

There will be cases when an application gets to production without going through the automated pipeline, or when a misconfiguration results in a vulnerable application. These cases make discovery and testing of web applications in production critical.

3

TO DETECT AND PROTECT AGAINST AN ATTACK.

Operations needs visibility into potential security issues in deployed software so that they can drive a quick response.

Having the Conversation

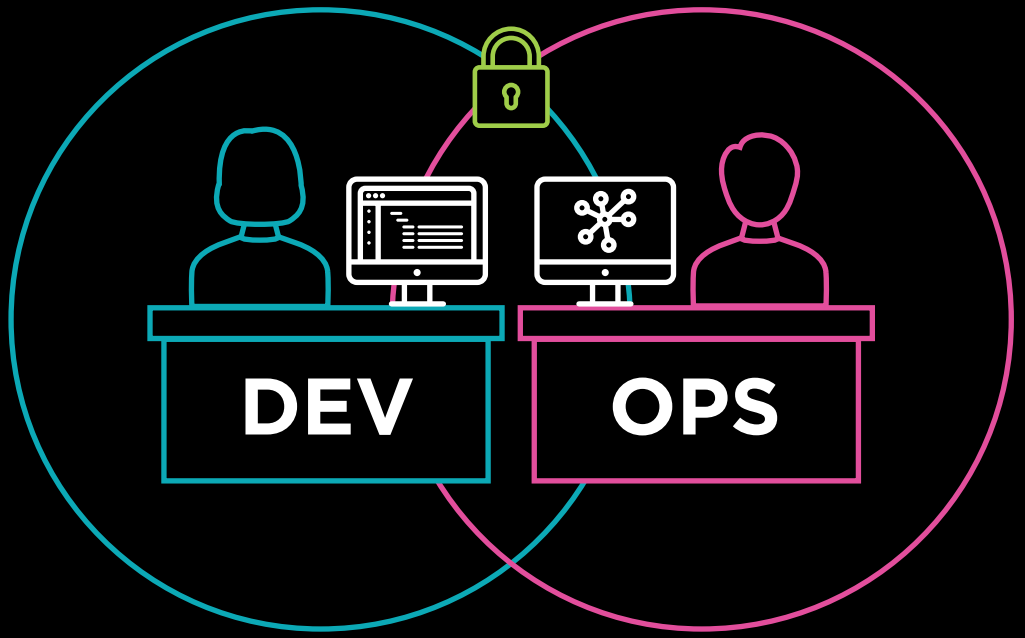
Questions to Ask When Integrating Security Into DevOps

Many organizations are at the earliest stages of considering how to integrate security into their DevOps practices. The following questions will help you think about how to design an integrated solution for securing the CI/CD pipeline:

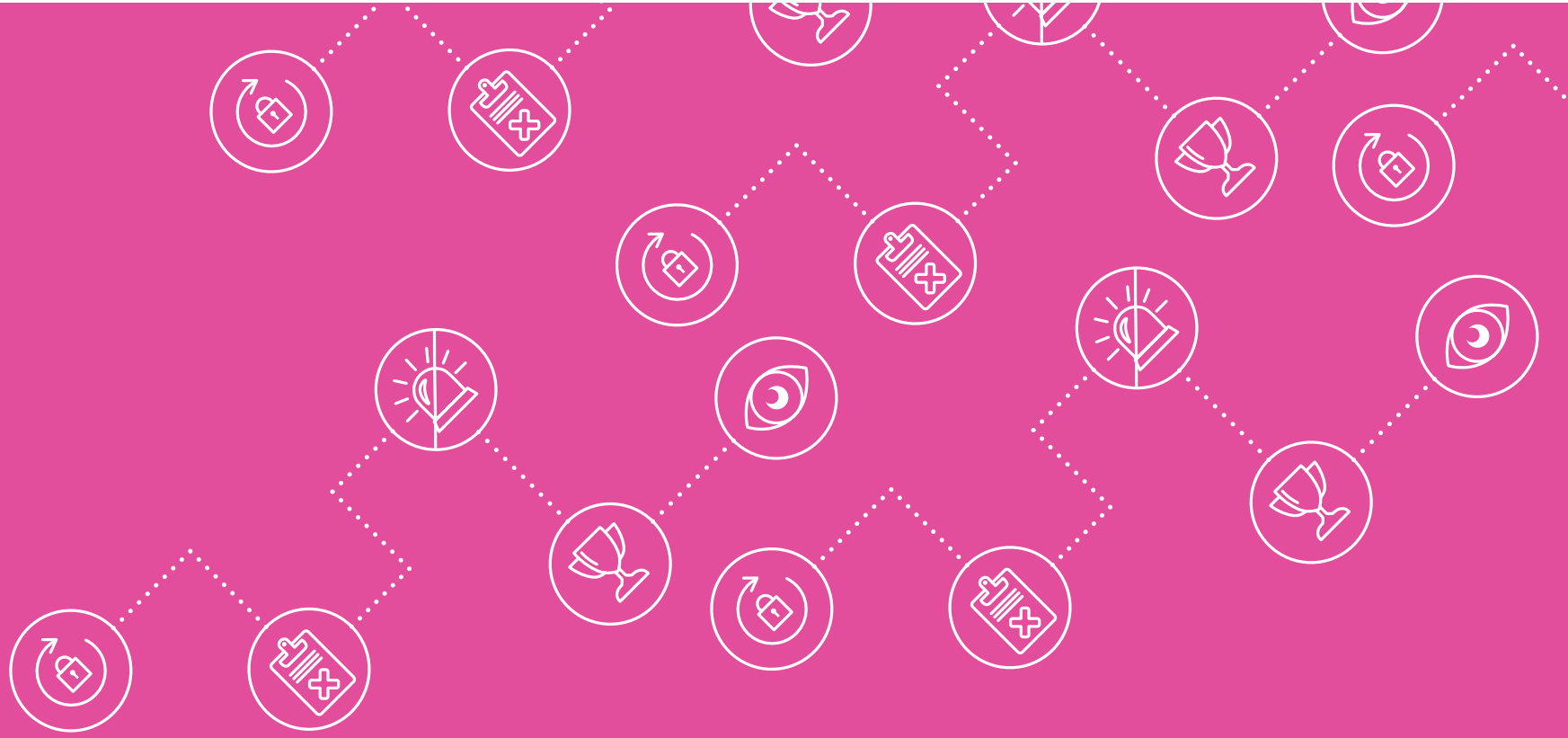
1. Have you rearchitected your applications for microservices, or is that work still in progress?
2. Which of your applications will pass through a CI/CD pipeline? Microservice-based? Monoliths? In what languages?
3. What tolerance do you have for "false alarms" (FPs) from an application security capability that is integrated into your DevOps practices?
4. Are you practicing trunk-based development, or do you still practice release and feature branching?
5. How do you plan to monitor your operational applications for security attacks?
6. How do you plan to bring security expertise into the DevOps team?

CONCLUSION

The process and technical requirements for integrating security with DevOps practices and CI/CD technology are challenging for any application security technology to meet.



By embracing DevOps principles and looking beyond the pipeline to organizational and production capabilities, you greatly increase the chances of successfully integrating security with DevOps.



VERACODE

SECURING THE SOFTWARE THAT POWERS YOUR WORLD.

Veracode's cloud-based service and systematic approach deliver a simpler and more scalable solution for reducing global application-layer risk across web, mobile and third-party applications. Recognized as a Gartner Magic Quadrant Leader since 2010, Veracode secures hundreds of the world's largest global enterprises, including 3 of the top 4 banks in the Fortune 100 and 20+ of Forbes' 100 Most Valuable Brands.

LEARN MORE AT WWW.VERACODE.COM, ON THE VERACODE BLOG, AND ON TWITTER.