digital.ai

2025 EBOOK

Shift Left into the Secure Quality Fast Lane

Embed Quality and Security from the Start to Accelerate Web and Mobile App Development

Table of Contents

What is Shift Left?

The Evolution of Software Development

The Testing Bottleneck

Enter Shift Left

The Shift Left Continuum

Shift Left in Practice

Organizational Impact of Shift Left

Tools and Enablers for Shift Left

Shifting Security Left: The DevSecOps Imperative

Summary

What is Shift Left?

Over the past few years, Shift Left has become a strategic imperative in modern application development. The primary driver is the growing expectation of users, making it increasingly important to deliver consistent and engaging experiences. This is driven by providing customers with consistent web and mobile applications across all channels. At the same time, development organizations are facing renewed pressure to deliver these applications while balancing speed with the desired quality and security.

Development organizations are facing renewed pressure to deliver these applications while balancing speed with the desired quality and security.

The Evolution of Software Development

Since the late 1990s, software development has been spurred on by the rise of the internet, and the ubiquitous integration of technology. This has led to a desire to continually shorten delivery cycles.

In the early 2000s, new methodologies began to emerge that accelerated development cycles. These began with Agile and DevOps but expanded to include Continuous Integration (CI) and Continuous Delivery (CD), to name a few.

These practices share common principles like small incremental development, frequent code commits, and enhanced communication. All of this is designed with the desire to enable adaptability and early software validation.

The Cost of Error

As these new practices enabled agility and adaptability, a critical business challenge arose —namely, the ability to detect defects early. Research continues to show that the earlier functional, performance, accessibility, or security issues can be identified, the simpler and easier it is to resolve these issues.

The Testing Bottleneck

While Agile and DevOps were off accelerating development, they left one critical area behind: testing. This remained a sequential phase, creating a silo that often grew into a major bottleneck. This separation siloed testing, causing it to often grow into a significant bottleneck.

Testing was still performed manually, but as release cycles consistently shortened, it began to slow down the entire process. This was because paradoxically test volume was only partly tied to new functionality. The vast majority were regression tests that ensured existing features remained unbroken.

Another challenge born of this process was the misalignment of ownership and goals. Developers whose primary focus is on delivering code are not measured on code quality or 'automation-friendliness". We believe in the inherent strive for quality, but looming deadlines often relegate testing to a secondary concern.

This is the exact moment where Shift Left intervenes.

Enter Shift Left

To Shift Left is to integrate **quality, testing, and proactive application hardening** much earlier in the development cycle. This will ensure that robust security mechanisms are embedded as early as the design and build phases. This approach shares responsibility for application delivery among a larger group, extending beyond just developers and test engineers.

Shift Left goes beyond simply changing release times. It implies a fundamental shift in mindset, accountability, organizational processes, and structure.

Development organizations are facing renewed pressure to deliver these applications while balancing speed with the desired quality and security.

The Shift Left Continuum

Organizational shift-left maturity varies greatly. The spectrum includes initial integrations that lead up to a comprehensive 'shift further left' into design and requirements.

There are, however, some similarities that define the continuum:

Growing emphasis on automation across all stages is essential. This includes embedding hardening mechanisms during the build phase and ensuring unhindered testing of these hardened applications.

Increasing collaboration comes with shared ownership of processes and projects across development, quality, and security teams.

Focusing on integrating quality and security by incorporating Test-Driven Development (TDD) and security principles that build resilience into the app itself.

Shift Left in Practice

Implementing Shift Left, regardless of the specific flavor, necessitates major changes. As you introduce quality, app hardening, and automation early in the process, it impacts:

- Roles and responsibilities
- Skill sets
- Processes
- Infrastructure and tools

Organizational Impact of Shift Left

As the focus on early automation expands to include **application hardening and self-protection,** the organization will see a significant shift in resource and skills. The transformation moves away from reliance on manual testing and turns toward integrating automation to engineer quality and security.

Traditionally, we See Resource Allocation Focus:

10% Performance & Volume:

- Manual performance testing, basic volume checks.

20% Environment & Data:

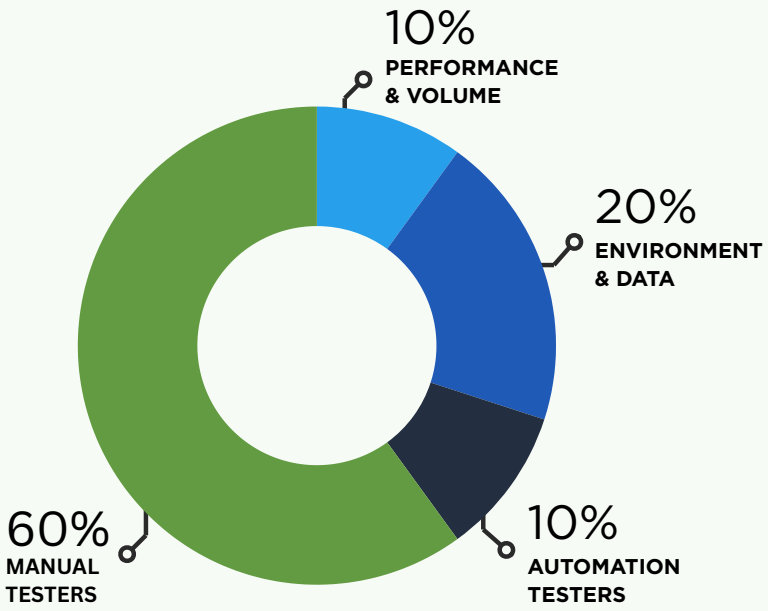
- Manual environment setup, data preparation.

10% Automation Testers:

- Limited regression automation, basic service virtualization.

60% Manual Testers:

- Extensive manual system, integration, and acceptance testing.



With Shift Left, This Focus Becomes:

5% Performance & Volume:

- Automated performance and volume validation.

10% Environment & Data:

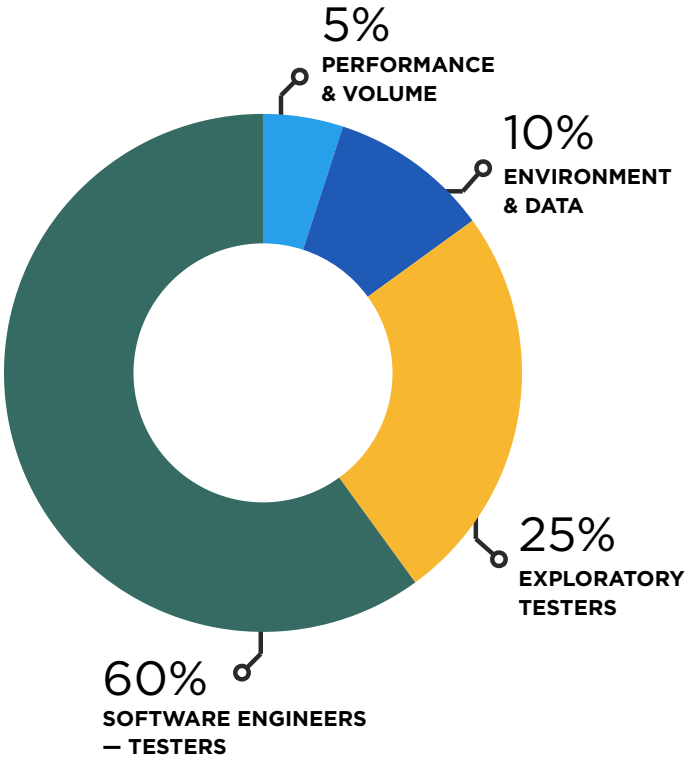
- Automated environment provisioning, data management.

25% Exploratory Testers:

- Deep domain expertise, advanced business rules testing, user-centric acceptance testing.

60% Software Engineers — Testers:

- Driving automation roadmap, continuous testing, **integrating application hardening and RASP during build, ensuring automated testing of hardened apps without interference,** user story thinning, and advanced service virtualization.



The following table shows the effect shift left has on different roles and what they can do to get ready for the change:

ROLE	KEY CHANGE	WHAT YOU NEED TO DO
Developer	Focus expands beyond code to the entire product. Responsible for embedding quality and security by design and automating unit tests.	• Learn automation principles for easily testable code. • Understand core quality assurance (QA) methodologies. • Grasp secure coding practices during the build phase.
Manual Tester	Traditional use case testing shifts toward becoming a domain expert, customer champion, and exploratory testing specialist.	• Decide whether to specialize as a domain expert or transition into automation. a. Domain experts must deepen product business knowledge, enhance communication with developers, and cultivate soft skills. b. Automation engineers must learn a coding language (no need to be an expert programmer).
Automation Engineer	Role expands from just writing tests to enabling large-scale quality and security automation.	• Deepen coding knowledge and readability to write stable, efficient tests. • Learn about Continuous Integration (CI) and parallel execution. • Master cross-platform automation (web, mobile). • Understand application hardening to ensure tests validate functionality without triggering anti-tamper protections.

Tools and Enablers for Shift Left: A Secure, Unified Digital Quality Lab

A unified testing solution that encompasses browsers, mobile devices, simulators, and emulators while enabling large-scale parallel execution is critical. Such a solution helps to:

- Achieve scale and speed for quality on secured apps.
- Ensure uniform testing processes across digital platforms.
- Share and optimize resources to promote efficiency.
- Share and reuse test cases between developers and testers.
- Drive quick and valuable feedback.
- Consolidate test results, enabling analytics and actionable insights.

An Integrated Security and Quality Toolchain

Software development has undergone a paradigm shift, moving away from monolithic systems. The order of the day is flexible, integrated toolchains. With fast-paced cultures like DevOps, Agile, and CI/CD taking over, teams need specialized tools tailored to their specific functions.

For instance, teams should use a test execution environment capable of running tests developed in various languages and frameworks. This allows developers to use frameworks like XCUITest or Espresso, while automation engineers might prefer Appium. The uniqueness of Digital.ai is that this can all occur while integrating application hardening and RASP during the build phase. A well-integrated toolchain prevents manual intervention, reduces errors, and speeds up the development cycle. Opting for open tools that adapt and integrate easily will streamline handovers, especially when embedding critical security capabilities.

Summary

We exist in a world where the relentless demand for faster web and mobile app releases continuously drives paradigm shifts in development. Methodologies like Agile, DevOps, and CI/CD have altered the SDLC dramatically. At the same time, quality and security have struggled to keep pace. To ensure speed does not compromise application integrity, we value a shift-left mentality that integrates comprehensive quality and proactive security hardening much earlier in the development timeline.

Embracing this shift is more than just a matter of timing. It demands fundamental changes in organizational structure, tooling, and methodologies. The challenge is worth the effort as leveraging Digital.ai Testing enables automated embedding of security and unhindered testing of hardened applications. We can help every organization find its optimal "Shift Left" sweet spot, unlocking substantial value in speed, quality, and resilience.

To ensure speed does not compromise application integrity, we value a shift-left mentality that integrates comprehensive quality and proactive security hardening much earlier in the development timeline.

The Digital.ai Difference

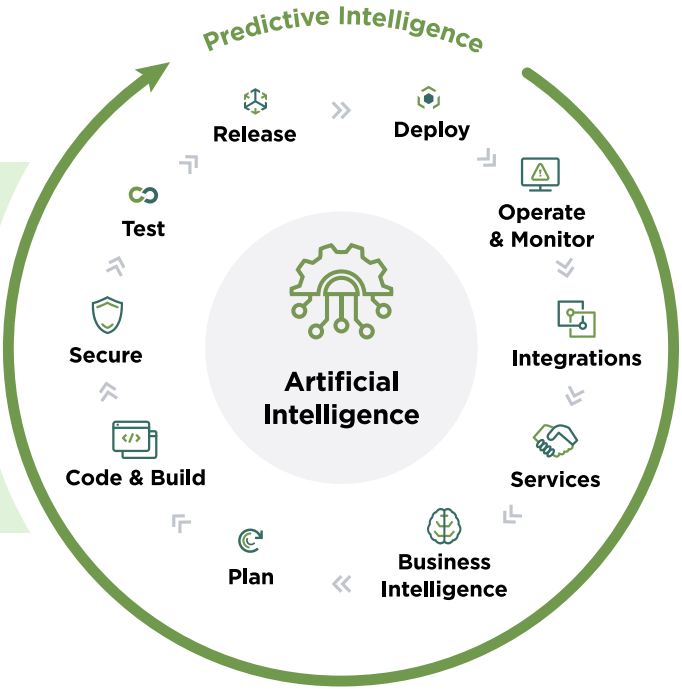
One platform that automates release pipelines and integrates complex toolchains.
Unify app delivery, integrate existing tools & scale across any environment.

Automated mobile app testing and security designed to scale.
Provide secure, high-quality apps through better automated testing & app protection techniques.

Built-in AI, intelligence, compliance, and governance across all software delivery workflows.
Centralize data, optimize processes, and gain augmented insights for faster, safer software delivery.

Digital.ai AI-powered DevSecOps platform

Helping you harmonize software delivery



About Digital.ai

Digital.ai is the only AI-powered software delivery platform purpose-built for the enterprise, enabling the world's largest organizations to build, test, secure, and deliver high-quality software. By unifying AI-driven insights, automation, and security across the software development lifecycle, Digital.ai empowers enterprises to deliver innovation with confidence. Trusted by global 5,000 enterprises, Digital.ai is redefining how enterprises build better software in an AI-driven world.

Additional information about Digital.ai can be found at digital.ai/ and on Twitter, LinkedIn and YouTube.

Learn more at Digital.ai

digital.ai™

©2025 Digital.ai is a trademark of Digital.ai Software, Inc.