

Original Article

# Beyond Deployment: Unveiling the Dynamics of Shift-Right Testing

Parvathy Purushothaman

Principal QA Developer, UKG, Canada .

Corresponding Author : [parvathy.purushothaman@ukg.com](mailto:parvathy.purushothaman@ukg.com)

Received: 17 December 2023

Revised: 23 January 2024

Accepted: 07 February 2024

Published: 19 February 2024

**Abstract** - This paper investigates the latest approach in testing, known as Shift-Right testing, within the context of modern software development. Diverging from the established Shift-Left approach, Shift-Right testing prioritizes post-deployment testing, with a specific focus on real-world scenarios in the live production environment, thereby departing from conventional testing practices. This paper delves into the key characteristics, benefits, tools, and practices associated with Shift-Right testing. This also highlights how shift left testing aligns with the principles of continuous improvement, user-centric design, and efficient incident response. Finally, this document also discusses how the integration of Shift-Right testing with Agile methodologies can accelerate time-to-market, enhance collaboration, and facilitate risk mitigation. In conclusion, the paper strongly recommends the widespread incorporation of Shift-Right testing in modern software development to meet the changing needs for software quality and user satisfaction.

**Keywords** - Shift-right testing, Post-production testing, Continuous improvements, User-centric development, Real-world validation.

## 1. Introduction

In the rapidly evolving landscape of software development, the quest for impeccable application quality and reliability stands as a paramount objective for development teams. Embracing early testing methodologies has emerged as a cornerstone strategy, reflecting teams' recognition of the imperative to unearth bugs and issues at the nascent stages of development. This proactive testing approach has not only proven invaluable but has also yielded tangible benefits in terms of heightened overall quality and elevated customer satisfaction levels.

Yet, despite the meticulous efforts invested by development and testing teams, the deployment of applications often unveils unforeseen challenges when encountered by real users. These post-deployment hurdles, rooted in the intricate and multifaceted nature of real-world environments, pose distinctive obstacles that conventional testing methods may fail to anticipate or address fully. Navigating these complexities inherent in real environments presents a formidable task, with a plethora of configuration intricacies and nuances that defy seamless replication in testing phases. Consequently, users may encounter unanticipated issues or "cracks" within the application fabric that remained undetected during the testing regime. The ensuing scramble to rectify these discrepancies underscores the critical significance of post-deployment testing in

safeguarding user satisfaction and maintaining the integrity of the application ecosystem.

Enter shift-right testing—a strategic paradigm shift that refocuses attention on monitoring and evaluating application performance post-deployment. Diverging from traditional testing methodologies fixated on pre-launch assessments, shift-right testing extends the evaluation continuum beyond deployment, placing a premium on real-world usage patterns and ongoing monitoring. By keenly observing user interactions within their native environments, shift-right testing empowers teams to identify and remediate emergent issues swiftly, ensuring seamless user experiences and sustained application robustness.

### 1.1. Shift Left Vs. Shift Right

Shift Left emphasizes the early detection of issues in the product life cycle through practices such as UX reviews, code reviews, static code analysis, and unit testing. Conversely, Shift Right places its focus on ongoing monitoring in the live environment, utilizing tools like APM, infrastructure monitoring, and RUM to ensure continuous evaluation and improvement. Shift left is often associated with Test Driven Development (TDD) and continuous integration, whereas shift right aligns more with DevOps, continuous testing, feedback, and adaptation. Both approaches together can contribute to a comprehensive and effective testing strategy in modern software development.



## 2. Key Characteristics of Shift-Right Testing

### 2.1. Post-Production Focus

Shift-Right testing places a primary focus on testing activities in the production environment. This strategic emphasis enables the simulation of real-world conditions, allowing for the identification of issues that may only manifest in the live operational setting. By extending testing into the post-production phase, organizations ensure a comprehensive understanding of how the software behaves under actual usage conditions.

### 2.2. User Feedback

Shift-Right testing revolves around obtaining continuous user feedback and seamlessly integrating it into development cycles. This process provides insights into various scenarios that may not be evident otherwise. By leveraging real user feedback, this approach enhances overall usability, customer satisfaction, and loyalty, ensuring the software aligns closely with user expectations.

### 2.3. Performance Monitoring

Shift-Right testing continuously monitors the application's performance and behavior in production. This helps in identifying and addressing performance bottlenecks and other critical issues promptly and efficiently.

### 2.4. Seamless Integration with DevOps

Shift-Right testing seamlessly integrates with DevOps, leveraging automation and collaboration to embed testing in the continuous delivery pipeline. Tests run at key stages like code commit and deployment, catching issues early and improving software quality. While automation is vital, teamwork among developers, testers, and ops engineers remains essential for agile and resilient software delivery.

### 2.5. Experimentation

Shift-Right testing encourages experimentation and A/B testing in the live environment. This provides valuable insights into user behavior and preferences and equips organizations to make informed, data-driven decisions.

## 3. Benefits of Shift-Right Testing

### 3.1. Real-World Validation

Shift-Right testing ensures software performs as expected in the live production environment, providing authentic validation under actual conditions. This approach captures issues not evident in pre-production, offering a realistic assessment considering user interactions and system load.

### 3.2. User-Centric Development

Focusing on user interactions, Shift-Right testing aligns development with user expectations. This user-centric approach, enriched by real user feedback, elevates the overall user experience, fostering increased satisfaction and loyalty.

### 3.3. Proactive Issue Identification

By observing real-world usage patterns and promptly detecting any anomalies or issues, this methodology enhances the product's resiliency and stability. Not only does this proactive approach lead to an improved user experience, but it also safeguards the organizational reputation by enabling swift responses to and resolution of potential problems.

### 3.4. Efficient Incident Response

Shift-Right testing's proactive issue identification leads to quicker incident responses, minimizing user impact. This efficiency enables rapid bug fixes or improvements, maintaining a positive user experience and organizational reputation.

### 3.5. Continuous Improvement

Insights derived from post-production testing in Shift-Right contribute to continuous improvement, optimizing functionality and performance over time. This iterative process aligns with changing user needs and evolving market demands, ensuring the software remains dynamic and responsive.

### 3.6. Cost Savings

Detecting issues early in Shift-Right testing reduces emergency response costs, preserving resources. This proactive approach prevents widespread user dissatisfaction, and the efficient incident response further safeguards the organization's reputation.

### 3.7. Adaptability to Changes

Shift-Right testing's proactive identification enables swift adaptation to changes, whether induced by updates, increased user traffic, or external factors. This flexibility ensures the software remains resilient in dynamic environments.

## 4. Tools for Shift Right Testing

### 4.1. Application Performance Monitoring (APM) Tools

#### 4.1.1. New Relic

Track essential metrics, visualize dependencies, and promptly identify issues through alerts and error tracking. Gain insights into critical transactions, conduct synthetic checks, and monitor browser performance for comprehensive observability.

#### 4.1.2. AppDynamics

Monitors application performance, business transactions, and infrastructure in real time and provides advanced diagnostics and analytics.

#### 4.1.3. Dynatrace

AI-driven APM tool that automatically detects and analyzes performance issues. Offers insights into user experience, application dependencies, and infrastructure.

## **4.2. Infrastructure Monitoring Tools**

### **4.2.1. Datadog**

Datadog provides metrics, visualizations, and alerts to ensure your engineering teams can maintain, optimize, and secure your cloud or hybrid environments.

### **4.2.2. Nagios**

Open-source monitoring tools for systems, networks, and infrastructure allow for customizable monitoring solutions through plugins.

### **4.2.3. Prometheus**

An open-source systems monitoring and alerting toolkit was originally built at SoundCloud.

## **4.3. Real User Monitoring (RUM) tools**

### **4.3.1. New Relic**

Provides real-time insights into user interactions, page load times, and overall application performance. It helps identify bottlenecks and optimize the user experience.

### **4.3.2. AppDynamics**

Monitor user interactions, track performance, and identify issues impacting the end-user experience. It offers visibility into both frontend and backend performance.

### **4.3.3. SpeedCurve**

Specializes in web performance monitoring and RUM. It tracks user-centric performance metrics, visualizes performance data over time, and helps teams optimize their web applications.

### **4.3.4. Akamai mPulse**

Monitor user interactions, track performance, and gain insights into how users are experiencing web applications. It helps organizations optimize content delivery.

## **4.4. Error Tracking Tools**

### **4.4.1. Sentry**

Captures and reports errors in real-time and provides detailed information about errors, including stack traces and affected users.

### **4.4.2. Rollbar**

Monitors errors and exceptions in web and mobile applications and offers real-time error tracking and notifications.

## **4.5. User Feedback and Analysis Tools**

### **4.5.1. UserVoice**

It enables organizations to collect, prioritize, and analyze user suggestions. It allows users to submit ideas, and it includes features for voting and commenting on proposed enhancements.

### **4.5.2. Qualtrics**

Experience management platform that offers survey and feedback capabilities. It allows organizations to design and distribute surveys, analyze responses, and derive insights.

### **4.5.3. Hotjar**

It helps organizations understand how users interact with their websites, identify pain points, and gather feedback through targeted surveys using heat maps, recording sessions and surveys.

## **4.6. Log Analysis tools**

### **4.6.1. ELK Stack (Elasticsearch, Logstash, Kibana)**

A powerful combination of log analysis and visualization. Elasticsearch indexes and stores logs, Logstash processes and structures the logs, and Kibana provides a user interface for visualization.

### **4.6.2. Splunk**

Offers log analysis and monitoring capabilities to help in searching, monitoring, and analyzing machine-generated data.

## **5. How to Set Up Shift Right Testing Practice**

### **5.1. Focus on User-Centric Development**

Integrate User Feedback and Analysis tools throughout the development lifecycle. Conduct A/B testing and experiments using RUM tools to evaluate user experience impact. Align development efforts with real user expectations and preferences.

### **5.2. Establish Continuous Improvement Process**

Establish Continuous Improvement Process Leverage insights from APM, RUM, User Feedback, and Log Analysis tools to drive continuous improvement. Establish effective feedback loops connecting development, operations, and user feedback channels. Utilize continuous monitoring to dynamically adapt to shifting user needs and market demands.

### **5.3. Implement Proactive Issue Identification and Resolution in Production**

Set up a centralized dashboard for instant updates on application health, performance, user behavior, infrastructure status, errors, and feedback. Set up proactive alerting mechanisms to swiftly detect and address anomalies. Collaborate with Product Managers for effective issue triage and priority assignment, dedicating around at least 10% of the team's velocity to ensure dedicated resources for issue resolution.

### **5.4. Foster a Culture of Data-Informed Decision-Making**

Nurture a team culture centered around data-driven decision-making, emphasizing the importance of informed choices based on real-time insights from diverse sources. Encourage the comprehensive integration of metrics derived from various monitoring and feedback tools, establishing a

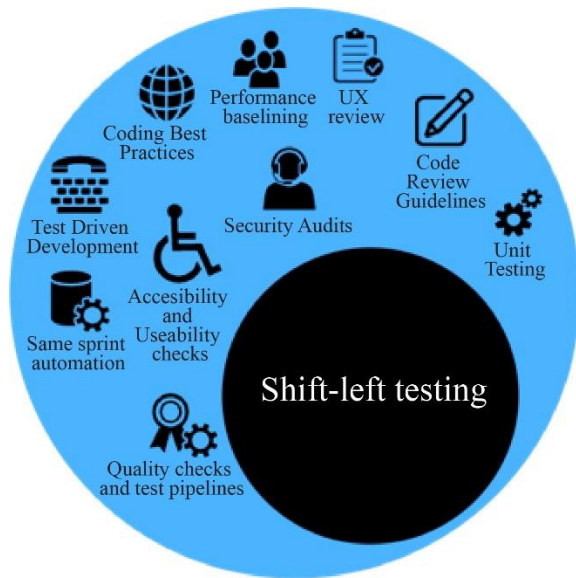
robust foundation for effective and well-rounded decision-making processes.

### 5.5. Develop an Efficient Incident Response

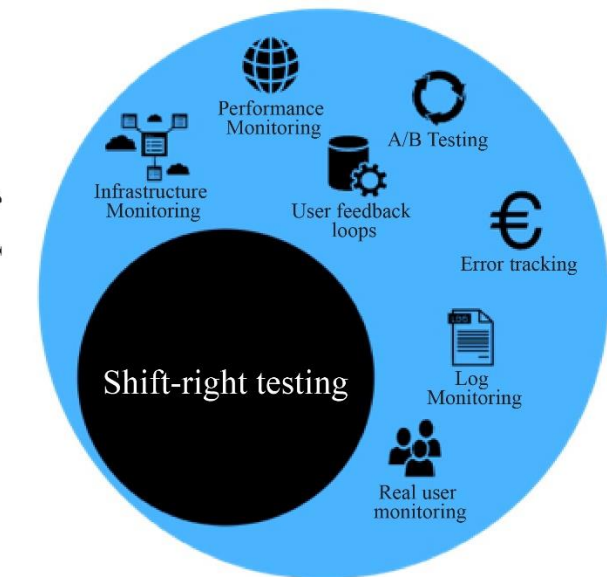
Establish a streamlined incident response mechanism by leveraging alerts from diverse monitoring and tracking tools. Minimize user impact through prompt resolution of identified issues. Implement on-call procedures and schedules to ensure timely responses to incidents. Develop detailed playbooks outlining defined steps and designated points of contact for each incident type, enhancing the efficiency and effectiveness of incident response efforts.

## 6. Shift Right Testing and Agile

Shift-Right testing strongly complements Agile methodologies, seamlessly aligning with their core principles. Its continuous feedback not only offers valuable insights into user interactions but also facilitates quick adaptation to changes, a central aspect of Agile methodology. Enhancing Agile's user-centric development focus, Shift-Right testing concentrates on real user experiences, contributing to heightened user satisfaction. Furthermore, it supports Agile's dedication to continuous improvement, utilizing post-production insights to optimize functionality and performance



Production Deployment



## 8. Conclusion

In summary, Shift-Right testing is a strategic methodology focusing on post-deployment testing, contrasting with Shift-Left's early testing. This method, emphasizing real-world scenarios and continuous monitoring, offers a unique approach to ensure software resilience and user satisfaction. It not only complements Agile methodologies but

also extends beyond functional accuracy to encompass usability, reliability, and adaptability. The proactive issue identification, continuous improvement, and cost-saving benefits make Shift-Right testing indispensable. By integrating various tools and fostering a data-driven culture, organizations can efficiently implement and enhance their Shift-Right testing practices, ensuring software excellence in today's evolving environment.

## 7. Blending Shift-Right and Shift-Left Testing: A Complete Testing Strategy

Combining both Shift-Right and Shift-Left testing methodologies forms an all-encompassing strategy for software testing.

Shift-Left testing focuses on early detection and prevention of defects during the product lifecycle, involving activities such as UX review, unit testing, code reviews, test automation, performance benchmarking, etc. This allows teams to identify and address issues at the earliest stages of development, reducing the risk of larger problems later on. Shift-Right testing, on the other hand, occurs post-deployment and involves monitoring the software in real-world environments. This includes gathering user feedback, analyzing performance metrics, and monitoring logs to promptly identify and resolve any issues that may arise in production. By merging these two approaches, organizations gain a holistic view of product quality.



## References

- [1] What is Application Performance Management (APM)?, Cisco, 2024. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/cloud/what-is-apm.html>
- [2] Dig into New Relic APM 360, New Relic, 2024. [Online]. Available: <https://newrelic.com/platform/application-monitoring>
- [3] Smart Agent for Cisco AppDynamics, AppDynamics, 2023. [Online]. Available: <https://www.appdynamics.com/>
- [4] Unified Observability and Security, Dynatrace, 2024. [Online]. Available: <https://www.dynatrace.com/>
- [5] Modern Infrastructure Monitoring, Datadog, 2024. [Online]. Available: <https://www.datadoghq.com/product/infrastructure-monitoring/>
- [6] The Industry Standard in IT Infrastructure Monitoring, Nagios, 2023. [Online]. Available: <https://www.nagios.com/>
- [7] What is Prometheus?, Prometheus Authors, 2024. [Online]. Available: <https://prometheus.io/docs/introduction/overview/>
- [8] Mpulse, Akamai, 2024. [Online]. Available: <https://www.akamai.com/products/mpulse-real-user-monitoring>
- [9] Code Breaks, Fix it Faster, Sentry, 2024. [Online]. Available: <https://sentry.io/welcome/>
- [10] Discover, Predict, and Resolve Errors in Real-Time, Rollbar, 2024. [Online]. Available: <https://rollbar.com/>
- [11] Understand What Your Customers Really Want, UserVoice, 2024. [Online]. Available: <https://www.uservoice.com/product/collect-user-feedback>
- [12] Make Every Interaction an Experience that Matters, Qualtrics, 2024. [Online]. Available: <https://www.qualtrics.com/>
- [13] Everything You Ever Wanted to Know About Your Website, Hotjar, 2023. [Online]. Available: <https://www.hotjar.com/>
- [14] Meet the Search Platform that Helps You Search, Solve, and Succeed, Elastic Stack, 2024. [Online]. Available: <https://www.elastic.co/elastic-stack/>
- [15] Ensure Your Enterprise is Unstoppably Resilient, Splunk, 2024. [Online]. Available: <https://www.splunk.com/>