



Sri Lanka Institute of Information Technology

**SOFTWARE ENGINEERING PROCESS
& QUALITY MANAGEMENT**

**Emerging Trends in Software Testing
Assignment 01**

Y3S2-WE-34

	Student ID	Student Name
1	IT22364388	Peiris P.V. G
2	IT22915740	Kulathunga K.A.K.M
3	IT22306104	Gunarathne M.D.C.H
4	IT22342744	Dissanayake E.G.M

Table of Contents

Introduction.....	3
Identifying a Novel Trend in Software Testing.....	3
Overview of the Trend	3
Reasons for its Emergence	4
Practical Implementation Strategy.....	5
Plan to integrate the trend into a Real-World project.....	5
Tools & Technologies.....	5
Implementation Roadmap	5
Challenges & Mitigation Strategies	6
Challenges	6
Mitigation Strategies	6
Perspectives on Feasibility	6
Key Quality Metrics for Evaluating Modern Testing Trends.....	7
Industry Adoption & Agile/DevOps Impact.....	7
Real world Examples	7
Netflix – Shift-Right Testing & Chaos Engineering	7
Amazon – Shift-Left Testing & Security Compliance.....	8
Integration with Agile & DevOps	8
Diagram of Shift-Left & Shift-Right in CI/CD.....	9
Comparative Analysis & Challenges	9
Comparison with Traditional Testing.....	9
Advantages and limitations Shift-Left Testing:	10
Case studies or industry Use Cases	10
Future Predictions & Ethical Considerations.....	11
Future Predictions (Next 3-5 Years).....	11
Ethical Considerations	11
Interview with an Industry Expert.....	12
References:.....	12

Introduction

Ensure reliable and secure apps – software testing is essential for making reliable and secure apps. The complexity and speed of software development sometimes hampers traditional testing approaches. Shift-left and shift-right testing are two modern testing strategies that combined testing in different stages of software development life cycle. Shift-Right helps in providing greater monitoring and resilience in real scenarios whereas Shift-Left involves early-stage testing. The latter improves the generation of flaws, while the former helps in prioritizing early steps testing. For this purpose, a case study of the real world, case studies of Google and Uber will be presented. These cases will be compared with the existing common test implementations. Risks, issues, and mitigations will be discussed along with future perspectives for AI-powered testing and data privacy issues.

Identifying a Novel Trend in Software Testing

Overview of the Trend

Shift-Left & Shift-Right Testing: Concepts and Importance

Software testing has become matured by implementing Shift-Left and Shift-Right testing techniques. Both these methodologies contributed to fine-tuning the quality of software through testing at various levels of the Software Development Life Cycle (SDLC). Strategically scheduling the testing activities can help organizations improve software reliability, reduce delivery timelines, and optimize investments

Shift-Left Testing: Definition and Concept

Shift Left testing is shifting testing phases into an earlier stage in the software development life cycle. For the most part, testing phases are done at the end of development, which is why defects are normally found too late, and fixing them costs a lot. Shift-Left testing thus enables the initial stages of development in the company to concentrate on detecting as many defects as possible as soon as possible for better quality and lower cost of operations.

Key Aspects of Shift-Left Testing:

- Unit and Integration Testing Early: Developers create and run unit tests while they are coding.
- Static Code Analysis: SonarQube and ESLint are applications that detect possible problems before executing the program.
- Automated Functional and Security Testing: Removes human errors and raises test coverage.
- Continuous Testing in CI/CD Pipelines: Automated testing is integrated into automated pipelines to detect problems before release.

Shift-Right Testing: Definition and Concept

Shift right testing, however, is performed in real world production environments after the software has been released. This approach focuses on performance, monitoring, usage, and failure testing, to name a few. It guarantees that the applications are working as they are expected to in real world conditions and that any issues are found and fixed quickly.

Key Aspects of Shift-Right Testing:

- Real-time Monitoring & Observability: New Relic and Dynatrace are tools that provide application performance insights and identify anomalies.

- A/B Testing and Feature Flags: Launch features to real users in controlled batches and get feedback.
- Chaos Engineering: Gremlin and Chaos Monkey are platforms that inject controlled chaos to test the resilience of the system.
- Performance & Security Testing in Production: To check for system stability during peak traffic and to avoid breaches.

Reasons for its Emergence

Shift Left and Shift Right testing techniques have gained popularity due to the widespread adoption of Agile and DevOps methodologies. Companies' fast-paced release cycles and the need for higher software quality have resulted in a demand for continuous testing approaches.

1. Market Demand for Faster Software Delivery

- Businesses need to release software at a fast speed to be in the game in shifting markets.
- Agile development approaches are characterized by shorter lead time between the delivery of production iterations and hence require some preliminary and continuous testing.
- Shift-Left allows premises diagnosis of delays in production at early detections in time.
- Shift-right allows real-time performance evaluation and monitoring for live applications.

2. Advancements in Technology

- Test Automation: For example, Jest, JUnit, Selenium, etc. automate unit, integration, and functional tests.
- Monitoring & Observability: some of the tools like a new relic, Datadog, Splunk, that assist with very deep level performance analysis.
- Chaos Engineering: We can observe companies such as Netflix running controlled failures to test the robustness of a system.
- AI & Machine Learning: AI based testing solutions can foresee failures and optimize test cases automatically.

3. Cost & Time Efficiency

- Defect Prevention vs. Defect Detection: Fixing bugs in the early stages (Shift-Left) is significantly cheaper than addressing them in production.
- Reduced Downtime: Shift-Right testing helps detect and fix production issues before they cause major disruptions.
- Better Resource Allocation: Organizations can optimize testing efforts by balancing Shift-Left and Shift-Right approaches.
- Higher ROI: Organizations that implement these testing methodologies see improved product quality and faster releases.

Real-World Adoption Examples:

1. **Netflix:**
 - Uses Shift-Right testing extensively with Chaos Engineering to simulate failures and ensure high availability.
 - Implement feature toggles to roll out features gradually and analyze user feedback.
2. **Amazon:**

- Follows Shift-Left security testing, integrating vulnerability scans into CI/CD pipelines to catch issues early.
 - Uses canary releases to ensure new updates do not negatively impact user experience.
3. Google:
- Shift-Left testing is applied in security and unit testing to detect coding errors before deployment.
 - Shift-Right testing with real-time telemetry and AI-based failure detection helps optimize production environments.

Practical Implementation Strategy

Plan to integrate the trend into a Real-World project

Combining Shift-Right and Shift-Left By ensuring early issue recognizable proof and continuous observing, testing into a real-world extend makes strides computer program quality. Whereas Shift-Right ensures real-time execution examination in generation, Shift-Left concentrates on proactive testing all through advancement. To make a solid, successful testing technique all through the SDLC, this arrange addresses issues like computerization overhead and security concerns whereas sketching out the devices, stages, and best hones vital for smooth usage.

Tools & Technologies

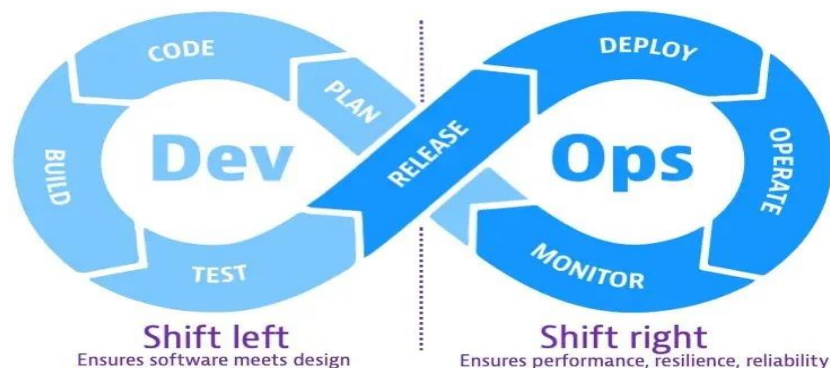
Shift-Left Testing Tools:

- JUnit, Selenium: Utilized for early-stage unit and integration testing.
- SonarQube: Static code examination apparatus that makes a difference distinguish issues early within the development handle.
- Checkmarx, Veracode: Security testing tool coordinates early within the SDLC
- GitLab CI/CD, Jenkins: Continuous Integration tools that bolster testing during the development lineup.

Shift-Right Testing Tools:

- Chaos Monkey (Netflix): Helps test systems in production by randomly shutting down instances, guaranteeing strength in production.
- New Relic, Datadog: Monitoring tools to track application execution in generation and recognize issues.
- Sentry: Real-time error tracking for applications to distinguish bugs and post-deployment issues.
- LoadRunner, Apache JMeter: Performance testing tools used in production-like situations.

Implementation Roadmap



Challenges & Mitigation Strategies

Challenges

- **Tool Integration:** Combine Shift-Left tools into existing CI/CD pipelines can be challenging.
- **Cultural Shift:** Developers and analyzers may not be usual to focus on testing early (Shift-Left) or post-deployment observing (Shift-Right).
- **Performance Overhead:** Implementing Shift-Right tools like chaos engineering can include execution overhead.
- **Security Corners:** Testing in production (Shift-Right) can uncover vulnerabilities in live system framework.

Mitigation Strategies

- **Tool Training:** Train developers and analyzers (testers) utilize the proper devices and coordinate them viably into the workflow.
- **Automate Everything:** Automate testing, monitoring, and feedback loops to ensure continuous quality.
- **Fail-Safe Mechanisms:** Implement rollback methodologies, such as feature flags, to moderate risks when testing in generation.
- **Collaboration Between Groups:** Empower collaboration between development, operations, and QA groups to grasp both Shift-Left and Shift-Right procedures.

Perspectives on Feasibility

When considering the feasibility of embracing Shift-Left and Shift-Right testing, organizations must evaluate the benefits, challenges, and practical viewpoints of implementations.

1. Feasibility Factors for Shift-Left Testing

Pros:

- **Early Imperfection Discovery-** Finding bugs early on means less costly fixes down the line.
- **Improve Code Qualities –** Consistent testing leads to superior coding habits.
- **Faster Development Cycle -** integration tests and Automated unit make releases happen more rapidly.

Challenges:

- **Starting Learning Bend-** Developers must be talented in testing to form their work.
- **CI/CD Complexity -** Setting up mechanized (automated) pipelines requires specialized knowledge.
- **Test Maintenance Overhead -** Tests require regular upgrades to keep up with frequent code changes.

2. Feasibility Factors for Shift-Right Testing

Pros:

- **Testing in the Real World -** This method lets you reveal execution and security issues as they happen in real utilization scenarios.
- **Boosting Client Fulfillment -** By gathering criticism and monitoring in real-time, you'll be able to make the software better and more enjoyable for clients.
- **Greater Dependability-** Utilizing chaos engineering techniques makes the system more robust and way better prepared to handle unexpected issues.

Challenges:

- **Potential for Live System Glitches -** Testing on live frameworks (systems) continuously carries the risk of service interruptions or blackouts.

- Higher Cost Tag - You'll have to spend money on monitoring tools and possibly on cloud situations for testing.
- Require for a Solid DevOps Mentality - Success relies heavily on near collaboration between developers, operations staff, and the QA group.

Key Quality Measurements for Evaluating Modern Testing Trends

Shift-Left Metrics:

- Code Scope: Percentage of code tested by unit tests and integration tests
- Deformity Density: The number of defects found amid early testing stages.
- Pass Rate Test: Percentage of tests that pass in CI/CD pipelines.
- Mean Time to Detect (MTTD): How rapid defects are identified during developments.

Shift-Right Metrics:

- Production Occurrence Frequency: Number of incidents or issues that happen within the production environment.
- Customer Fulfillment: Criticism from users based on performance, usability, and functionality after deployment.
- Mean Time to Recovery (MTTR): How long does it take to settle (fix) issues that arise in production.
- Service Availability: Uptime or accessibility of the application in production.

Industry Adoption & Agile/DevOps Impact

Shift-left and shift-right tests are widely utilized in top technological companies to improve software quality. Underneath are two large companies, **Netflix** and **Amazon**, and how they have utilized these test techniques.

Real world Examples

Netflix – Shift-Right Testing & Chaos Engineering

Netflix is a pioneer in Shift-Right testing, particularly with its Chaos Engineering approach. This strategy guarantees that their platform is very exceptionally adaptable despite the errors. It is mentioned here how Netflix's shift-right applies:

1. Chaos Engineering with Chaos Monkey
 - Netflix deliberately presents errors in production using a device called Caus Monkey.
 - This closes the server randomly to test the system's capacity to overcome unexpected errors.
 - This helps Netflix distinguish shortcomings and improve fault resistance.
2. A/B Testing for Feature Rollouts
 - Netflix uses A/B testing to slowly roll out modern features.
 - They release new features to a small percentage of clients, monitor their behavior, and after that choose whether to expand the release.
 - This decreases risks and guarantees only successful changes reach the complete audience.
3. Observability & Real-Time Monitoring

- Netflix uses tools like New Relic and Prometheus to monitor system wellbeing.
- These tools help identify errors, execution bottlenecks, and unusual user behavior in real-time.

Amazon – Shift-Left Testing & Security Compliance

Amazon integrates Shift-Left testing in its development life cycle to ensure performance, compliance and security.

1. Automated Security Testing Early in Development
 - Amazon implements Shift-Left security testing using SonarQube and Checkmarx.
 - These tools analyze code during development to detect early security holes.
2. Integrated continuously with Unit & API Testing
 - Amazon uses unit tests, API testing, and integrated tests in its CI/CD pipelines.
 - Tools like Postman, JUnit, and Selenium ensure that new changes do not break the existing features.
3. Real-Time Monitoring & Response in real-time
 - AWS CloudWatch and AWS X-Ray are used for real-time monitoring of AWS services.
 - Amazon's Shift-Right testing helps them collect logs and monitor problems after deployment.

Integration with Agile & DevOps

Shift-left and shift-right testing play an important role in development and agile work processes, providing continuous tests, faster releases and high software quality.

Shift-Left Testing in Agile:

- Agile development follows a repeated sprint cycle, which means developers often publish new features.
- Shift-Left testing helps in Agile by:
- Early detection of errors: Developers write unit tests and perform static code analysis in each sprint.
- Test-Driven Development (TDD): Ensures that code passes tests before deployment, reducing bugs in later stages.
- Continuous Feedback Loop: Test results are immediately checked, improving the quality of the software.

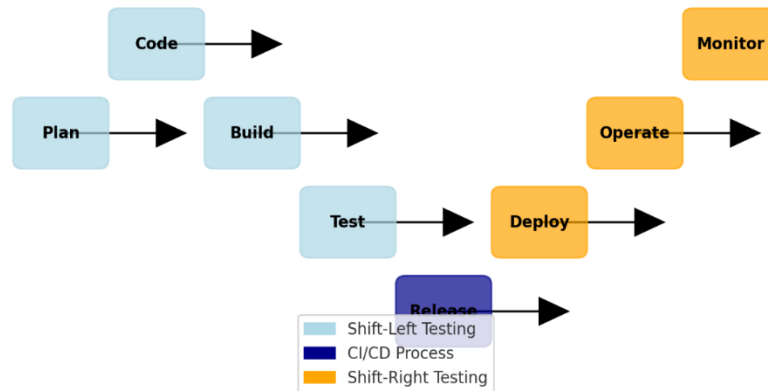
Shift-Right Testing in DevOps:

DevOps focuses on automation and continuous implementation. Shift-Right testing ensures quality after deployment through:

- Live Monitoring & Observability: Tools such as Prometheus, Datadog, and New Relic track performance in production.
- Canary Releases: Small portions of users get the new update first, allowing quick return in case of problems.
- Chaos Engineering: Helps groups actively check failure in the real production environment.

Diagram of Shift-Left & Shift-Right in CI/CD

Modified DevOps Lifecycle with Shift-Left & Shift-Right Testing



1) Shift-Left Testing (Light Blue):

- Including Plan, Code, Build, and Test phases.
- Focus on early-stage testing to catch defects before deployment.

2) CI/CD Process (Dark Blue):

- The Release phase marks the conversion from development to deployment.

3) Shift-Right Testing (Orange):

- Including Deploy, Operate, and monitoring.
- Focus on post-deployment testing such as observability, chaos engineering, and performance monitoring.

Continuous Testing in CI/CD Pipelines: Automatically testing is integrated into automated pipelines to detect problems before release.

Comparative Analysis & Challenges

Comparison with Traditional Testing

Traditional Testing:

- This is conducted at the end of the software development cycle.
- Mainly focusing on identifying and correcting errors before the release version.
- Rely strongly on manual testing. (Limited automation)
- These errors are only treated after detection. (Reactive approach)

Shift-Left Testing:

- Often in the requirements and design stages of the SDLC, integration from the beginning.
- Mainly aims at preventing defects by detecting them and fixing them as early as possible.
- Early defect detection reduces future risks. (Proactive approach)
- Focusing on AI-centralized tools and integrate continuously. (Highly automated)

Shift-Right Testing

- Implemented after implementation in the production environment.
- Focus on monitoring and improving system reliability under real-world conditions.
- Improve stability and user experience and allow for immediate risk response.
- Use real -time observation and monitoring tools for continuous feedback. (Datadog, New Relic, Prometheus etc.)

Advantages and limitations

Shift-Left Testing:

- **Advantages:**
 - Reducing the cost and effort of fixes using early detection of errors.
 - Improve collaboration between Testing groups and developments.
 - By integrating testing into the development process, it improves the quality of the software.
- **Limitations:**
 - Requires a cultural shift and training for development groups to adopt testing responsibilities.
 - Increase the workload and complexity due to the initial implementation.

Shift-Right Testing:

- **Advantages:**
 - Validates software performance and reliability in the real-world scenarios.
 - Provides insights into the user behavior and system usage patterns.
- **Limitations:**
 - The potential risks of exposing users to production defects.
 - Analysis and management of large quantities of production data.

Case studies or industry Use Cases

Google:

Google uses Shift-Left testing by including security evaluations at the beginning of the development process, ensuring that vulnerabilities are resolved before release. At the same time, they deploy Shift-Right testing through continuous performance monitoring and A/B testing in production, allowing the time to optimize the actual time depending on the commitment of the user.

Uber:

Uber has implemented Shift-Left testing through the company scale program known as BITS (Backend Integration Testing Strategy), which enables on-demand deployment and routing to test sandboxes. This method allows individual commitments to be verified concurrently before integration, ensuring the separation between production and testing traffic. Uber also uses chaos engineering approaches as part of its Shift-Right testing strategy to ensure the recovery of the system during the peak traffic periods.

Risk and Mitigation Strategies

Risks in Implementation:

- **Overhead in Automation:** Implementing significant automation might result in expanded complexity and maintenance requirements.
- **Data Privacy Concerns in Shift-Right Testing:** Testing in production systems may uncover sensitive user data, raising security and privacy concerns.
- **Resistance to Cultural Change:** Shifting testing roles may meet pushbacks from groups used to standard strategies.

Mitigation Strategies:

- **Specific Test Automation:** Prioritize automation for repeated and high-risk test cases to adjust the advantages with the maintenance overhead.
- **Data Anonymization and Encryption:** Implement solid data anonymization and encryption components to defend client security and privacy during production testing.
- **Comprehensive Preparing and Communication:** Conduct training sessions and keep open communication channels to empower the cultural transition towards coordinates testing procedures.

Future Predictions & Ethical Considerations

Future Predictions (Next 3-5 Years)

1. **AI-Driven Testing:**

- By facilitating intelligent test automation, predictive analytics, and self-healing frameworks, artificial intelligence (AI) will improve both shift-left and shift-right testing.

2. **Autonomous Testing Pipelines:**

- The AI integration will result in independent testing pipelines that can optimize test scope and adjust to changes within the code of little assistance from humans.

3. **Improved Predictive Analytics:**

- Shift-Right testing will utilize AI to anticipate conceivable system malfunctions and performance snags before they influence clients, empowering prompt fixes.

4. **Security Enhancements:**

- To ensure that vulnerabilities are settled all through the SDLC, there will be greater emphasis on incorporating security testing into both Shift-Left and Shift-Right methods.

Ethical Considerations

Privacy and Security Concerns in Shift-Right Testing:

- ❖ **Ethical Dilemma:** Regulations and client privacy may be violated when client data is gathered and analyzed in production settings.

- **Solution:** Utilize data anonymization procedures and make sure that data protection laws just like the GDPR are followed.

Interview with an Industry Expert

- ❖ Interview Link:
<https://drive.google.com/drive/folders/10WQKjs1N9bVJ0NMmzladM48xVDB4qX5H?usp=sharing>
- ❖ Expert's LinkedIn Profile Link: https://www.linkedin.com/in/lumini-nanayakkara-6b7886215?utm_source=share&utm_campaign=share_via&utm_content=profile&utm_medium=ios_app

Individual Contributions:

	Student ID	Student Name	Individual Contribution
1	IT22364388	Peiris P.V. G	<ul style="list-style-type: none"> Identifying a Novel Trend in Software Testing
2	IT22915740	Kulathunga K.A.K.M	<ul style="list-style-type: none"> Industry Adoption & Agile/DevOps Impact
3	IT22306104	Gunarathne M.D.C.H	<ul style="list-style-type: none"> Comparative Analysis & Challenges Future Predictions & Ethical Considerations
4	IT22342744	Dissanayake E.G.M	<ul style="list-style-type: none"> Practical Implementation Strategy Interview mainly Host by

References:

(Mohan, 2025)

(Shift-Left vs Shift-Right Testing Approaches for Manual Testers, Sep10, 2024)