Matillion
Quality in Test Strategy

November 2025

Table of contents

1. Introduction

This document serves as a high level guide to our Quality in Test (QiT) strategy at Matillion, ensuring world class quality in our services. It outlines our testing approach, standards, tools, and validation processes. This document serves as an abstracted version of the detailed internal QiT strategy.

2. Key Points

Scope: Our focus is on quality in testing, ensuring adherence to standards outlined in this document.

Environments: We maintain various environments provisioned via Infrastructure as Code (IaC) using Terraform, allowing us to promote changes into various test environments.

Test Approach: We adopt a proactive testing approach aligned with our shift-left culture.

Quality Gates: Automated validation of quality gates at every pipeline stage ensures code integrity, backed with exploratory testing.

Pyramid: Our testing pyramid emphasises efficient testing with cheaper tests at the base and fewer expensive end-to-end tests.

Test Types: Various testing types are identified and described throughout the Software Development Life Cycle (SDLC) described below.

Tools: We use a variety of tools for effective testing throughout the SDLC. Our tools support security, service level, performance, scanning and e2e testing.

Metrics: Transparent reporting of quality metrics ensures accountability and continuous improvement.

Security: Security is integrated into the DNA of our designs and supplemented with thorough testing.

Performance: We continuously validate service and application performance across three core areas. Service level, integration level, E2E level.

Test Standards: Detailed requirements for testing standards cover best practices and specific testing types. The standards are baked into frameworks and documentation.

Bugs and Incidents: We have a zero-bug policy and clear processes for handling bugs and incidents.

Quality Community: Our quality community drives knowledge sharing and best practices. Incident Lifecycle: Clear incident management processes ensure timely resolution and minimise customer impact.

Risk: We balance risk and reward in our engineering practices through comprehensive risk assessment.

Team Agreement: Teams commit to a binding contract outlining their responsibilities for maintaining testing standards.

Implementation Guide: Structured guidelines help teams adopt and implement our testing standards effectively.

3. Validation Process Details

Each service must undergo the following quality validation steps prior to deploying to production:

- PRR Compliance (Production ready)
- On-call Compliance
- Dispatch Integration (Controlled deployments)
- Security Scans
- Code Scan
- Unit Tests
- Integration Tests
- Component Tests
- Contract Tests
- Can I Deploy Validation
- Service Level Deployment Tests
- End-to-End (E2E) Smoke Tests
- End-to-End (E2E) User Journey Tests
- Pipeline Execution (Validating internal components)
- Exploratory testing

4. Additional Considerations

- **Synthetic monitors** are deployed on each service.
- **Deadlock Scenario Handling** is validated in mocked incident rooms.
- Every change is promoted and deployed through test environments prior to reaching customer facing environments.

- All features and traffic is controlled through **feature flagging**.
- Environments are configured in **multiple regions with integrated contingency**.
- Continuous validation of each environment is handled through a **central QE team**.

This comprehensive validation process ensures that each service meets high-quality standards and is ready for deployment with minimal risk.

5. Document Revisions

Date	Author	Description
November 2025	Steve Warburton	**First publication** • Initial release of the document. • Formatting and layout set for readability.