

# QA Process Proposal – Shopify Merchant Platform

**Prepared by:** Brian Muraya

**Role:** QA Engineer

**Date:** November 2025

## 1. Overview

This proposal outlines a lean QA process designed for a fast-moving Agile startup supporting **Shopify merchants**. The goal is to ensure reliable, secure, and merchant-friendly releases without slowing development speed.

## 2. QA Objectives

- Detect defects early through integrated QA in every sprint.
- Maintain confidence in rapid releases via test automation.
- Validate key user journeys and API reliability for Shopify integrations.
- Provide fast, data-driven feedback to developers.

## 3. QA Approach

### a. Manual Testing

Manual testing will handle new features, UX validation, and exploratory testing.

- **Focus areas:** Functional, UI/UX, regression, and cross-browser testing.
- **Tools:** TestRail (test cases), Jira (bug tracking).
- **Outcome:** Clear documentation of steps, expected results, and severity for every defect.

### b. Automated Testing

Automation will cover high-impact and repetitive flows such as merchant login, product sync, and checkout.

- **Tools:** Playwright or Cypress for UI; GitHub Actions/Jenkins for CI/CD.

- **Strategy:** Automate regression and smoke suites triggered on every pull request.
- **Benefit:** Faster feedback and reduced manual workload.

### c. API Testing

API testing ensures reliable communication between Shopify and internal services.

- **Tools:** Postman/Newman for endpoint validation.
- **Coverage:** Authentication, order updates, and inventory APIs.
- **Automation:** Included in CI/CD pipelines to verify backend stability before UI testing.

## 4. QA in the Agile Cycle

1. **Sprint Planning:** QA reviews user stories and defines acceptance criteria.
2. **Mid-Sprint:** Test as features are developed in staging.
3. **End of Sprint:** Run automated regression and smoke tests before release.
4. **Post-Release:** Monitor defect trends and validate production fixes.

## 5. Continuous Improvement

- Track automation coverage and prioritize high-risk areas.
- Share QA insights in retrospectives.
- Regularly review bug patterns for root cause prevention.

## 6. Expected Outcomes

- Faster feedback loops and safer releases.
- Balanced mix of manual insight and automation efficiency.
- Reliable, merchant-focused product quality aligned with Agile delivery speed.