# Final Assignment: Agile QA Test Strategy for Streamline Retail

coursera

## Updated Case Study

## Organization Overview

Streamline Retail is a mid-sized SaaS company that provides POS and inventory management solutions for small to medium-sized retailers. The company's products include a desktop management platform and a mobile app that store managers use for real-time inventory tracking, barcode scanning, sales dashboards, and analytics. It also has an optional web-store that the store managers may enable to showcase the stock or to sell products online.

## Recent Project

The company has recently released **Streamline Mobile App 2.0**, a complete rebuild of the mobile application. This project adopted an Agile Scrum framework to improve delivery cadence and quality. However, early metrics (test coverage, defect leakage, user adoption rate) have indicated lingering issues with quality, performance, and customer satisfaction.

## Current Initiative

The leadership team is now requesting a **comprehensive quality strategy** that sets a quality-first foundation for all future releases of the mobile app and related services. Your QA team has been tasked with producing this document.

## Your role

As a QA lead, you need to define a comprehensive quality strategy document.

# Hands-on Exercise

The task

Write a comprehensive **Test Strategy Document** for Streamline Retail’s mobile app and QA operations.

In should consist of 5 main parts:

1. Quality goals and standards
2. Process integration
3. Defect management process
4. Verification and validation
5. Continuous improvement

## Instructions

For this exercise make as many assumptions as you need in addition to the case study summary above. Feel free to consult external resources or find examples online to ignite creative thinking or get inspiration.

## 1. Quality Goals & Standards

Define what “quality” means for Streamline Retail.

Learnings from previous projects and best practices to consider:

- Product requirements should contain both requirements from internal stakeholders and external users’ expectations
- Alignment with business goals (e.g., customer satisfaction, usability, security).
- Quality metrics and acceptance thresholds (e.g., < 2/month defect leakage, ≥80% test coverage, 99.5% uptime).

## 2. Process Integration

Describe how QA integrates into each phase of the Agile development process. This section should include:

- QA roles in **Agile ceremonies** (Sprint Planning, Daily Standups, Sprint Reviews, Retrospectives).
- **Special QA activities and checkpoints** (e.g., creation of definition of done, test strategy sign off, three amigos, regression testing).

- Quality activities embedded in the **SDLC**, including:
  - Acceptance criteria definition
  - Unit testing
  - Code reviews
  - Test automation in CI/CD pipelines
  - Manual testing strategies for exploratory and usability testing
- A **RACI matrix** outlining roles and responsibilities for Developers, QA, Product Owner, and Scrum Master in quality activities.

## 3. Defect Management Process

Document how defects are reported, prioritized, and resolved:

- Propose severity and priority definitions (e.g. what makes highest severity different from low severity)
- Responsibility for triaging and responding to defects (roles and responsibilities between QA and other team members).
- SLA targets for defect resolution (MTTD, MTTR) based on severity, assuming customer satisfaction will drop if serious issues take longer than one business day to resolve; and lower priority issues take longer than a week to resolve

## 4. Verification & Validation

List and describe testing levels and types that will be used, including:

- Unit Testing
- Integration Testing
- End to end Testing
- User Acceptance Testing (UAT)
- Non-functional testing (Performance, Security, Usability)
- Preferred low-code/no-code automation tools
- Criteria for what will be automated and what will remain manual

## 5. Continuous Improvement

Describe how the QA team will continuously improve quality processes. Include:

- How retrospectives will be used to identify QA improvements.
- Feedback loops (e.g., bug root cause analysis, post-release reviews, user feedback).
- Quality-related KPIs that will be reviewed regularly.

## PAUSE HERE AND DON’T SCROLL UNTIL YOU’VE ATTEMPTED THE EXERCISE.

On the next page, there is the example solution. If you want to attempt this on your own, pause here and don’t scroll next until you are ready.

There are many different ways to accomplish this task, it is ok if your solution is different. Compare your solution with the one provided, if there are discrepancies think if you can explain them.

# Example Solution

Quality Strategy

1. **Quality Goals & Standards**

**Quality Definition:**
Quality at Streamline Retail means delivering secure, reliable, and user-friendly software that aligns with business needs and performs well under real-world retail conditions.

**Key Business Goals:**

Customer retention via exceptional usability. Our solutions will gain users' trust by being known for stability and exceptional security.

## Key Quality Goals:

- ≥80% unit and integration test coverage
- ≤ 2 reported production defects a month of severity higher than “low”
- ≥90% customer satisfaction rating
- 99.5% system uptime
- No critical security incidents in production

coursera

# 2. QA Process Integration

**QA in Agile Ceremonies:**

- **Sprint Planning:** QA estimates testing effort, plans test activities, and confirms test data requirements. QA reviews and refined acceptance criteria together with the business analyst and product owner.
- **Daily Standups:** QA shares testing progress, blockers, and syncs with engineers. Engineers provide proactive updates on upcoming functionality that will enter testing during that day.
- **Sprint Review:** QA demonstrates testing coverage, reports test results, and outstanding defects.
- **Retrospective:** QA's role is to suggest improvements to automation, process, and team collaboration.

## Embedded QA Activities:

- **Unit Testing** (Developer-led; 80% coverage)
- **Code Reviews** (Peer review including test review, developer-led)
- **End -to-end Test Automation** performed via low code automation (QA-led)
- **CI/CD** pipeline executes smoke, regression, and performance tests prior to every deployment
- **Exploratory Testing** in every sprint to catch edge cases and UX issues

coursera

## RACI matrix

| Activity | QA Engineer | Developer | Product Owner | Scrum Master |
|---|---|---|---|---|
| Write Unit Tests | C | RA | C | I |
| Define Acceptance Criteria | R | C | A | I |
| Execute Regression Tests | RA | C | I | I |
| Triage Defects | R | R | RA | I |
| Approve Release | C | C | A | R |

# 3. Defect Management Process

**Workflow:**

- Defects are logged in the task tracking tool (Jira).
- Triage occurs daily to assign priority
- QA assigns severity; Product Owner assigns priority.

**Severity Definitions:**

- **Critical (S1):**

## Acceptance criteria from the task definition are not met resulting in:

- The layout and overall visual appearance of the solution does not match the designs completely.
- Immediate affect on user's ability to execute primary scenarios, navigate the solution, or use its core functionality.
- Inability to perform major back-office functions such as customer support with no workaround available.
- Production issue having immediate affect on customer satisfaction or business continuity

## High (S2):

**Acceptance criteria from the task definition are partially met resulting in:**

- The layout and overall visual appearance of the solution match the designs, but there are considerable flaws including layout, accessibility or messaging.
- An issue having effect on user experience or back-office operation, which does not classify as S1
- An S1 defect that has a workaround.

**Medium (S3):**

Acceptance criteria from the task definition are almost met resulting in:

- minor visual defects are present, such as minor layout issues, insignificant functional flaws, or typos in messaging.
- an issue that does not materially affect the performance or functionality of the solution

**Low (S4):**

An issue raised is not covered by the acceptance criteria of any of the tasks, thus the issue can be considered an enhancement to the in-scope functionality and will be considered a change request.

coursera

## SLAs:

- **S1:** Fix within 1 day (with Mean time to detect aiming to < 1 hour)
- **S2:** Fix within 3 days (with Mean time to detect aiming to < 1 day)
- **S3:** Fix within 7 days unless deprioritized by Product owner
- **S4:** Move to Backlog for future prioritisation

## 4. Verification & Validation

For the purpose of establishing systemic quality practices the following conventions will be used:

- Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle: requirements are turned into very specific test cases, then the software is improved to pass the tests. This is opposed to software development that allows software to be added that is not proven to meet requirements.
- Behaviour-driven development (BDD) is an extension of test-driven development: development that makes use of a simple, domain-specific scripting language. It converts structured natural language requirements into executable tests. The result is a closer relationship to acceptance criteria for a given function and the tests used to validate that functionality.

3-amigos: agreement of scope of acceptance, alignment of amount of testing needed to acceptance, alignment and design of feature to meet outcomes

**The following verification and validation activities will be planned and executed:**

**1. Local development:**

- a. 3 amigos meeting to agree on testing scenarios and scope of automation
  - agreed scenarios documented in the task
- b. Unit tests creation
- c. Ad-hoc dev-qa catch ups for rapid validation

d. The development is not considered finished until

- i. the code is completed
- ii. unit tests are completed and pass
- iii. where required, services and libraries code test coverage is met 100%
- iv. where required, additional agreed automation is completed
- v. code review is passed
- vi. code is successfully merged into the higher level branch with automated regression passed

## 2. System integration testing

a. System integration test scenarios are updated when the requirements are finalised

b. System integration testing is performed by QA

c. Automation scenarios are automated as agreed

## 3. User acceptance testing

a. When the code is deployed to UAT environment, and end to end testing is performed by the QA

b. Test scenarios used for end to end testing are added to corresponding test suites, including regression test suite

c. Upon successful execution of end to end tests, a regression test suite is executed

d. Upon successful execution of regression test suite, an exploratory testing session is performed

e. Test automation tool is reconfigured to automate agreed scenarios

f. Once all of it is done, User acceptance testing is performed with the selected users or Product Owner as their delegate

## 4. Non-functional testing:

a. A specialised security partner will be employed to perform penetration testing, security scan, and performance testing. It will be done once per increment for a release candidate version of the product.

b. Usability testing will commence with the first visual prototypes being created and the results of testing will be incorporated in the requirements development phase of the project.

## Approach to automation:

1. 100% of unit tests and integration tests will be automated
2. End to end test cases will be automated as practical as per the following criteria:
   For a test case to be automated, it must have all of the following:
   a. Show high repetition
   b. Be otherwise time consuming to test manually
   c. Have algorithmically deterministic outcomes
   d. Be based on stable requirements

## OR at least one of the following:

e. Rely on large data sets

f. Involve cross platform testing

g. Have UI elements

## 5. Continuous Improvement

- Sprint Retrospectives will include QA insights review as a mandatory step (e.g., test failures, missed scenarios).
- Post-release Reviews will be performed to analyze defect leakage and user complaints.
- Regular bug root cause analysis will be employed to identify coding/test gaps.
- Review QA KPIs monthly (coverage, cycle time, leakage).
- Provide training and certification (all project members will be trained in TDD, BDD, test automation tools).

## Instructions for Documenting and Sharing The Project for Peer Review

### 1. Document the Project

- Use word processing software to create your report.
- Ensure all sections of the project document structure are completed.
- Proofread and edit your report for clarity and accuracy.

### 2. Share the Project

- Save your report and presentation in PDF format.
- Upload your documents to the “Peer Review” area in the course shell.
- Provide a brief description of your project in the submission post.

# Instructions for Documenting and Sharing The Project for Peer Review

3. Peer Review
   - Review the projects submitted by your peers.
   - Provide constructive feedback and comments on their work.