

CASE STUDY

Swag Labs — Solo Manual QA Cycle

Test planning · Manual execution · Negative/boundary testing · Release recommendation

Overview

A self-directed manual QA cycle on a Sauce Demo-style e-commerce storefront, run end to end as a portfolio exercise: I wrote the test plan, designed the test cases, executed them by hand, logged every defect found, and closed the cycle with a release recommendation — the same lifecycle a professional QA engineer owns on a real project, minus the team.

My Role

Sole tester across all roles: test planner, executor, defect logger, and reviewer. I deliberately kept the process disciplined rather than informal — writing a real test plan with entry/exit criteria and severity definitions before touching the site — because the goal was to practice professional QA rigor, not just “find some bugs.”

Results at a Glance

Metric	Result
Test cases executed	44
Pass rate	81.8% (35 passed)
Defects logged	8 — all Open
High-severity defects	4
Release recommendation	Not Ready

The Finding That Mattered Most

An 81.8% pass rate looks fine on paper. It isn't, once you see where the failures land. The four High-severity defects sit directly on the core purchase funnel:

- Add to Cart fails silently for specific products, on both the Listing and Detail pages.
- Remove is unresponsive once an item has been added — no way to undo the action.
- Checkout is reachable from an empty cart with no warning.
- The Finish button doesn't respond — no order can ever be submitted.

Put together, a shopper can be blocked at nearly every stage of checkout: adding the item, correcting a mistake, and finally submitting. That's why the report recommends halting release despite the headline pass-rate number — a good percentage doesn't mean much if the failures are concentrated exactly where revenue happens.

Test Design Approach

Rather than testing purely by intuition, I used a mix of structured techniques so coverage wasn't accidental:

- Equivalence partitioning and boundary value analysis on the checkout form fields (Zip/Postal Code formats, name-length limits).
- Decision tables for combinations of filled/empty checkout fields and their expected validation outcomes.
- Error guessing around known e-commerce failure patterns — double submission, empty-cart checkout, whitespace handling.
- Exploratory sessions specifically around the checkout → confirmation transition, since scripted cases alone tend to miss state-transition bugs.

What I'd Do Differently Next Time

- Add automated regression coverage for the core funnel (Add to Cart → Checkout → Finish) so these four defects, once fixed, can be caught immediately if they regress.
- Get access to a second reviewer — the test plan flags “no second reviewer” as a real risk, and it's the one limitation I couldn't mitigate as a solo project.
- Formalize the checkout validation rules with a requirements doc before the next cycle, since I had to base expected behavior on standard e-commerce norms in their absence.

Skills Demonstrated

- End-to-end test planning: scope, strategy, entry/exit criteria, risk register
- Structured test design: equivalence partitioning, boundary value analysis, decision tables
- Manual functional, negative/boundary, and exploratory testing
- Defect triage and severity assessment tied to business impact (funnel-blocking vs. cosmetic)
- Release-readiness reporting and recommendation

Full artifacts: Test Plan and Defect Summary Report (Word) — available in the project repository.