02 – Sprint Execution Protocol (QA Playbook)

SVS (Oh-Dish Backoffice) – Startup Simulation

Document Type: Protocol

Prepared By: QA – SH (Representative Protocol/Report)

Version: 1.0

Date: February 14, 2026

Status: Draft (Portfolio Simulation)

Note: This is a representative artifact created to simulate real enterprise processes in a startup environment. Use with the Agile Enterprise Framework docs as supporting analysis.

Approvals

Role	Name	Signature	Date
Product Owner	_____	_____	_____
Engineering Lead	_____	_____	_____
QA Lead/Mentor	_____	_____	_____
Stakeholder/UAT Rep	_____	_____	_____

Revision History

Date	Version	Author	Description
February 14, 2026	1.0	QA – SH	Defines sprint-level QA execution steps, evidence, and escalation

Table of Contents

1. Purpose & Scope

This protocol defines how QA executes within each 2-week sprint for SVS (Oh-Dish Backoffice) startup simulation. It provides a repeatable day-to-day workflow, checklists, and evidence requirements aligned to Dev→QA→Staging→Prod.

2. Sprint Timeline (2-Week / 10 Working Days)

Day	Primary Delivery Activity	QA Activity / Deliverable
Day 1	Sprint Planning	Confirm DoR, risk callouts, test data needs; create test tasks
Day 2–4	Development	Story-level testing on QA env; early automation stubs for smoke
Day 5	Mid-sprint checkpoint	Regression trigger review; defect aging review; update risk index
Day 6–8	Integration	Cross-module validation; run TestCafe smoke nightly; stabilize flaky tests
Day 9	Sprint Review prep	Finalize evidence package; QA summary slide (1 pager)
Day 10	Review + Retro	Present quality report; record leakage and action items

3. QA Activities by Ceremony

Sprint Planning – QA Checklist:

- Verify acceptance criteria are testable (positive + negative).
- Identify dependencies (RBAC, payments gateway, supplier sync).
- Confirm environment availability and test data requirements.
- Add QA tasks: test cases, automation updates, data validation, evidence capture.
- Call out high-risk stories (Payments, Orders state machine, RBAC, Reporting).

Daily Standup – QA Updates:

- What I tested yesterday (stories/modules).
- What I will test today.
- Blockers (environment, data, dependencies).
- Defects requiring triage or hotfix decision.

4. Test Data & Environment Protocol

Area	Data Needed	Who Provides	Storage/Evidence
Users/Roles	Admin, Manager, Cashier accounts; varied permissions	QA + Dev	Seed script + credentials vault
POS	Menu items, taxes, discounts, edge pricing	Dev + QA	Seed dataset + screenshots
Payments	Gateway sandbox credentials; test cards; failure codes	DevOps/Dev	Gateway docs + run logs
Orders	Historical orders; scheduled orders; refunds	QA	DB snapshot + report exports
Reporting	Known expected totals; reconciliation samples	QA	Expected vs actual worksheets

5. Regression Triggers & Selection Rules

Trigger Events (run targeted regression immediately):

- Changes to tax/discount/total calculations
- Changes to payment gateway integration or retry logic
- Changes to RBAC permissions or middleware
- Changes to order status transitions
- Schema changes in MySQL affecting orders/payments/users
- Production hotfix merged

Selection Rules (what to include):

- Always include critical flows: Login → POS Checkout → Payment → Order Completion → Report Verification.
- Include impacted module suites + dependent module suites (e.g., Payments impacts Orders and Reports).
- Include negative tests for RBAC and payment failure modes when those areas change.

6. Automation Expectations (TestCafe Studio)

Automation categories:

- Smoke (PR gate): Login, Users list, POS load, Orders list
- Critical regression: POS totals, payment success/failure, order status transitions
- Staging validation: end-to-end reconciliation checks (basic)

Automation hygiene rules:

- Stable selectors (data-testid) preferred; avoid brittle XPaths.
- Flaky tests: quarantine with ticket; fix within 1 sprint.
- Reports stored per pipeline run with timestamp and build id.

7. Evidence Package Requirements (Per Sprint)

Minimum evidence artifacts to attach/commit under /reports/Sprint_##/:

- Sprint Quality Report (template)
- Test execution summary (manual)
- TestCafe reports + screenshots (failures)
- Defect summary export
- Risk index update + top risks
- Go/No-Go recommendation (if release candidate)

8. Escalation & Blocker Handling

Situation	Severity	Who to Notify	Expected Action
Payment charged but order missing	Critical	Dev lead + PO immediately	Stop release; open incident; reconciliation
RBAC bypass / unauthorized API access	Critical	Dev lead + PO immediately	Patch + security review
Staging down /	Major	DevOps + Dev lead	Restore env; replan

unstable > 4 hours			testing
Flaky automation blocks pipeline	Major	QA + Dev	Quarantine test; fix selectors; stabilize

9. Templates (Checklists)

Sprint Planning QA Checklist (copy/paste):

- [ ] AC testable + negative cases
- [ ] Dependencies identified
- [ ] Test data requirements documented
- [ ] Regression impact estimated
- [ ] Automation tasks created
- [ ] Risks logged (payments/RBAC/reporting)

Supporting Analysis References (Framework Docs)

- 01_Agile_Operating_Model_v3.docx – Operating cadence, teams, CI/CD gates
- 02_Epics_and_Features_v3.docx – Module scope and backlog structure
- 03_User_Stories_v3.docx – Sample story formats + acceptance criteria patterns
- 04_Kanban_Board_Structure_v3.docx – WIP/SLAs/flow metrics
- 05_Lean_Principles_Application_v3.docx – Lean practices + evidence artifacts
- 06_PI_Planning_Simulation_v3.docx – 6-sprint PI plan + dependency matrix
- 07_Quality_Governance_Model_v3.docx – Exit criteria + Go/No-Go matrix
- 08_Metrics_and_Reporting_v3.docx – KPI dictionary + charts