03 – Defect Triage Protocol

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 severity, SLAs, triage workflow, RCA and leakage tracking

Table of Contents

1. Purpose

Standardize how defects are logged, classified, prioritized, triaged, and verified across SVS modules.

2. Defect Intake Standards

- Title format: [Module] – [Short issue] – [Environment]
- Must include: steps to reproduce, expected vs actual, environment/build, evidence (screenshots/logs), impacted users/roles
- Link to story/requirement if applicable; mark regression yes/no
- For payments: include transaction id/correlation id and reconciliation evidence

3. Severity Model

Severity	Definition	Examples	Customer Impact	Release Rule
Critical	Revenue/security/legal/compliance impact; blocks core flow	Payment charged but order missing; RBAC bypass; allergy info incorrect; double-charge	High immediate harm	No-Go
Major	High functional impact; workaround exists	Order status cannot change; discounts misapply in edge case; reports totals wrong for one branch	Significant disruption	Decision
Minor	Low impact cosmetic/rare edge	UI alignment; minor text; non-blocking sort issue	Limited	Can defer

4. Triage Workflow

1. Step 1: Validate reproducibility in same environment.
2. Step 2: Classify severity (Critical/Major/Minor) with rationale.
3. Step 3: Identify impacted module(s) + downstream effects (Payments→Orders→Reports).
4. Step 4: Assign owner + target fix sprint.
5. Step 5: Decide release impact: Block/Hotfix/Defer.
6. Step 6: Verify fix in QA; re-run impacted regression; close with evidence.

5. SLAs and Aging Rules

Severity	Acknowledge	Fix Start	Verify/Close Target
Critical	<2 hours	<1 day	<2 days
Major	<1 day	<3 days	<5 days
Minor	<3 days	Scheduled	Scheduled

Aging rule: any Critical open > 24 hours triggers escalation to Engineering Lead + PO.

6. Root Cause & Leakage Tracking

Root cause categories:

- Requirements gap
- Design flaw
- Implementation bug
- Data issue
- Environment/config
- Automation gap

Leakage stages (where defect was found): Dev, QA, Staging/UAT, Production.

7. Communication & Escalation

Trigger	Notify	Meeting	Decision Output
Critical payment/RBAC/allergy defect	Eng Lead + PO + QA	Immediate triage	Hotfix/Block/Rollback
3+ critical defects in same module	Eng Lead + PO	Risk review	Scope reduction; stabilization sprint
Staging UAT blocker	PO + Stakeholders	UAT sync	Re-plan; defer scope

8. Defect Ticket Template (Copy/Paste)

Title: [Module] – [Issue] – [Env]

Build/Version: ___ Environment: Dev/QA/Staging

Role/User: ___ Severity: Critical/Major/Minor Regression: Yes/No

Steps to Reproduce: 1) … 2) … 3) …

Expected Result: …

Actual Result: …

Evidence: screenshots/logs/link

Impact: who/what is affected; downstream modules

Notes: correlation id / transaction id (payments); DB check results (if applicable)

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