

Regression Testing Plan Example

Enterprise Software System

1. Introduction

This document defines the plan for regression testing of the **Enterprise Software System** following each release or code change. Regression testing ensures that recent changes do not negatively impact the existing functionality of the application.

2. Objectives

- Verify that new code changes do not introduce defects into existing features.
- Ensure core business workflows continue to function as expected.
- Reduce the risk of production outages after deployment.

3. Scope

Regression testing covers:

- All critical business modules (e.g., Authentication, User Management, Order Processing, Reporting)
- Integration points with external services
- User interface core flows

Out of scope:

- Non-affected legacy modules
- Third-party tools' internal functionality

4. Test Strategy

- Automated regression suite runs on every deployment to staging and production environments.
- Manual execution for critical scenarios and flows not covered by automation.
- Smoke testing after regression to ensure stability.

5. Test Scenarios

Module	Test Scenario	Type	Status
Authentication	Valid/invalid login, password reset, session timeout	Automated	Ready
User Management	Create, edit, deactivate user, assign roles	Automated	Ready
Order Processing	New order, update, cancel, process payment	Manual & Automated	In Progress
Reporting	Run sales report, filter data, export	Manual	Ready

6. Entry & Exit Criteria

Entry Criteria:

- All code changes have been unit tested and merged to main branch.
- Test environment is available and stable.

Exit Criteria:

- All planned regression test cases executed.
- No critical/blocker bugs remain unresolved.

7. Resources

- QA Lead: Jane Smith
- Automation Engineer: Tom Lee
- Manual Testers: 2
- Test Environments: Staging, UAT

8. Schedule

- Regression Test Planning: June 10 - June 11
- Test Execution: June 12 - June 15
- Reporting & Closure: June 16

9. Risks & Mitigation

- Risk: Incomplete automation coverage.
Mitigation: Prioritize manual testing for critical flows.
- Risk: Environment downtime.
Mitigation: Have backup testing environments available.

10. Approval

- Prepared by: QA Lead
- Reviewed by: Project Manager
- Approved by: Head of Engineering