# Test Closure Report & Defect Management Process

This document outlines the process for preparing a professional Test Closure Report/Test Summary Report and managing defects throughout the software testing lifecycle. It includes examples, defect life cycle stages, and severity/priority classifications.

## 1. Test Closure Report / Test Summary Report

The Test Closure Report is prepared at the end of a testing cycle to formally sign off that testing activities are complete for the defined scope. It summarizes testing efforts, results, and any remaining issues.

\*\*Contents of a Test Closure Report:\*\*

- How testing was conducted (manual, automation, environments used).  
- Number of defects identified.  
- Defect IDs mapped to corresponding requirements.  
- Defects deferred to the next sprint, if any.  
- Status of defects (closed by developers, pending issues).  
- Sign-off from the Test Manager confirming all in-scope requirements are tested and approved.

\*\*Example:\*\*  
Project: Online Banking Portal  
Testing Type: System + Regression  
Total Defects Raised: 25  
Defects Closed: 23  
Defects Deferred: 2 (Minor UI enhancements – planned for Sprint 12)  
Defect-Requirement Mapping: Req05 → DEF-101, DEF-102  
Sign-off: Test Manager John Smith – Approved

## 2. Defect Definition & Examples

A defect is any deviation of the application from the specified requirements. Defects can be critical (showstoppers), major, minor, or cosmetic.

\*\*Examples:\*\*  
- \*\*Showstopper:\*\* Unable to launch the application URL – identified during smoke testing.  
- \*\*Critical:\*\* Application launches but key transaction functionality fails.  
- \*\*Cosmetic:\*\* Font style, size, or color inconsistencies.

## 3. Example Defect Reporting

\*\*Test Step:\*\* Launch the banking application URL.  
\*\*Expected Result:\*\* Application should launch successfully.  
\*\*Actual Result:\*\* Error 503 – Service Unavailable.  
\*\*Severity:\*\* High  
\*\*Priority:\*\* High  
\*\*Defect ID:\*\* DEF-201

## 4. Defect Life Cycle

The defect life cycle represents the stages a defect passes through from identification to closure:

1. \*\*NEW (Tester)\*\* → \*\*OPEN (Tester)\*\* → \*\*FIXED (Developer)\*\* → \*\*RETEST (Developer)\*\* → \*\*CLOSED (Tester)\*\*  
2. \*\*RETEST (Developer)\*\* → \*\*REOPEN (Tester)\*\* → \*\*RETEST (Developer)\*\* → \*\*CLOSED (Tester)\*\*  
3. \*\*NEW\*\* → \*\*OPEN\*\* → \*\*DUPLICATE (Developer)\*\* – Duplicate defects are closed to avoid redundancy.  
4. \*\*NEW\*\* → \*\*OPEN\*\* → \*\*DEFERRED\*\* – Fix postponed to a future release.  
5. \*\*NEW\*\* → \*\*OPEN\*\* → \*\*REJECTED\*\* – Not considered a defect.

## 5. Severity & Priority

\*\*Severity:\*\* Indicates the impact of the defect on the application.  
- High Severity – Blocks core functionality.  
- Medium Severity – Major functionality affected but workarounds available.  
- Low Severity – Minor impact, cosmetic or enhancement issues.

\*\*Priority:\*\* Indicates the urgency of fixing the defect.  
- High Priority – Must be fixed immediately.  
- Medium Priority – Can be fixed in the current sprint.  
- Low Priority – Can be fixed in future sprints.

\*\*Examples:\*\*  
- High Severity, High Priority – Login button not working.  
- High Severity, Low Priority – Rare crash in an unused feature.  
- Low Severity, High Priority – Company logo missing on homepage before launch.  
- Low Severity, Low Priority – Minor alignment issues.

## 6. Defect Categories

- \*\*Complex\*\* – Requires multiple modules or deep investigation.  
- \*\*Major\*\* – Significant impact but limited to specific modules.  
- \*\*Minor\*\* – Small issues, minimal impact on users.