eTech S.C

Bug Reporting Template

By

QA Department

June 2025

Document management.

Company	eTech S.C.
Document title	Bug Report Template
Date	June-10-2025
Document Type	Template
Version	1.0

Table of Contents

Summary

This document outlines the standardized protocol for reporting software defects in the Quality Assurance (QA) department. The primary goal of implementing a formalized bug reporting template is to improve the clarity, completeness, and traceability of software defects throughout the product life cycle. This standard supports better communication between QA, Development, and Product Management teams and enhances overall product quality by ensuring efficient defect resolution.

Introduction

Software defects are an inevitable part of development. However, the effectiveness with which they are reported, communicated, and resolved can significantly influence project timelines and quality. Lack of standardization often leads to ambiguity, misinterpretation, and delays in resolution. This document introduces a consistent format for bug reporting using a predefined template that captures all essential information needed for effective debugging and resolution.

3. Objective

3.1 General Objective

To establish and institutionalize a comprehensive, consistent, and efficient standard for reporting software defects that supports quality assurance and continuous improvement across all QA and development activities.

3.2 Specific Objectives

- ✓ To define a uniform and repeatable process for logging software defects.
- ✓ To enhance the quality and clarity of defect communication between technical and non-technical stakeholders.
- ✓ To ensure every bug report includes sufficient, actionable information to facilitate timely and effective resolution.
- ✓ To reduce time-to-resolution and prevent rework by providing structured defect reports.
- ✓ To maintain an auditable, traceable, and organized record of reported issues for metrics, reviews, and process refinement.

4. Scope

This standard applies to:

- ✓ All QA engineers, software testers, and developers involved in any stage of the software development and testing lifecycle.
- ✓ Any software product developed, maintained, or tested by the organization.
- ✓ Internal applications, client-facing platforms, third-party integrations, and APIs.

5. Bug Report Template Structure

The standardized bug report shall consist of the following mandatory sections:

5.1. **Title**

- ✓ A concise summary of the issue.
- ✓ Use action-oriented language and include the affected module if applicable.
- ✓ Example: “[Login Page] Login fails with 500 error using valid credentials”

5.2. Description

- ✓ A detailed explanation of the issue, including contextual background such as what was being tested, by whom, and when.
- ✓ Mention whether the issue is reproducible or intermittent.

5.3. Environment Details

- ✓ Include:
  - o Device/Platform (e.g., Windows 10, macOS 13)
  - o Browser/OS version
  - o Application version/build number
  - o Test environment (e.g., staging, production, UAT)

5.4. URL (if applicable)

- ✓ Direct link to the affected page or endpoint where the bug occurs.

5.5. Steps to Reproduce

- ✓ Numbered sequence of steps to replicate the issue.
- ✓ Keep each step simple, clear, and precise.

5.6. Actual Result

- ✓ What was observed after executing the above steps.
- ✓ Clearly state any error messages, failures, or incorrect behavior.

5.7. Expected Result

- ✓ Clearly define what the correct behavior should be under the same conditions.

5.8. Severity

- ✓ Severity: Impact on system functionality (e.g., Critical, Major, Minor)

5.9. Attachments

- ✓ Include relevant files:
  - Screenshots
  - Console logs
  - Stack traces
  - Video recordings
  - JSON requests/responses (for API bugs)

6. Usage Guidelines

- ✓ The bug report must be completed **immediately** after detecting an issue.
- ✓ Reports should be submitted in the company's designated tracking system (e.g., Jira, Azure DevOps, Bugzilla).
- ✓ Reporters must review their report for completeness and clarity before submission.
- ✓ All sections of the template must be filled unless explicitly marked as optional.
- ✓ Each bug should be reported **individually**. Multiple issues must not be bundled in a single report.

7. Review & Verification Process

Once a bug is reported:

- ✓ A QA Lead or designated team member reviews the submission for clarity and completeness.
- ✓ The development team assesses the validity and assigns it for fixing.
- ✓ Status updates (e.g., Open, In Progress, Fixed, Reopened, Closed) should be tracked and communicated.
- ✓ Regression testing must be performed after the fix is deployed.

8. Metrics and Reporting

To continuously improve our QA process, the following metrics will be tracked:

- ✓ Bug detection rate by tester
- ✓ Time to resolution
- ✓ Number of duplicate/invalid bugs
- ✓ Distribution by severity and priority
- ✓ Bug reopening rate

Regular reports will be shared with QA leadership and development teams.

9. Continuous Improvement

This standard is subject to bi-annual reviews. Feedback and suggestions for enhancement are welcomed and may be directed to the QA Department Lead. Any revisions will be version-controlled and communicated to all stakeholders.

10. Appendix

10.1. Example Bug Report

Title: Profile Page's Image upload fails with 400 error

Description: When attempting to upload a profile image, the server returns a 400 Bad Request error.
Environment: window 11, Chrome 113.1, Build 3.2.1, UAT

URL: https://app.company.com/profile

Steps to Reproduce:
1. Login to the system
2. Navigate to Profile page
3. Click on "Change Avatar"
4. Upload a .png image under 1MB
5. Click Save

Actual Result: Error 400 is shown, and image is not saved
Expected Result: Image should upload successfully and display
Severity: Major
Attachments: Screenshot_2025-06-17.png, dev-console-log.txt