



BUG REPORTING STANDARDS

TechRacine Solutions Academy

<https://techracine.com>

<https://academy.techracine.com>

Study Guide

Table of Contents

Introduction to Bug Reporting	3
Importance of Proper Bug Reporting	3
What is a Bug.....	3
Bug Life Cycle.....	3
Components of a Bug Report	4
Writing Effective Bug Titles	4
Steps to Reproduce	5
Expected Result vs Actual Result.....	5
Severity and Priority	5
Attaching Screenshots and Logs.....	5
Bug Reporting Tools	6
Common Types of Bugs	6
API and Backend Bugs	6
UI and UX Bugs	7
Security and Performance Bugs	7
Bug Reporting Best Practices.....	7
Common Mistakes to Avoid	8
Sample Bug Reports	8
Summary.....	8

Introduction to Bug Reporting

Bug reporting is the process of documenting software defects identified during testing or usage.

A good bug report helps developers understand, reproduce, and fix issues efficiently.

Clear bug reporting improves software quality and communication between testing and development teams.

Importance of Proper Bug Reporting

Proper bug reporting reduces confusion and development delays.

Well-documented bugs help developers reproduce issues quickly.

Professional bug reporting improves project quality and team collaboration.

What is a Bug

A bug is a defect or issue in software that causes incorrect behavior.

Bugs may affect functionality, performance, security, or user experience.

Software bugs can occur due to coding mistakes, logic errors, or unexpected scenarios.

Bug Life Cycle

The bug life cycle defines different stages of a defect.

Common statuses include New, Assigned, Open, Fixed, Retest, Verified, Closed, and Reopened.

Bug tracking systems help manage these stages efficiently.

Components of a Bug Report

A professional bug report generally includes:

1. Bug ID
2. Title
3. Module
4. Environment
5. Steps to Reproduce
6. Expected Result
7. Actual Result
8. Severity
9. Priority
10. Attachments

Writing Effective Bug Titles

Bug titles should clearly describe the issue.

Avoid vague titles such as 'Application Error'.

Good titles improve readability and tracking.

Examples:

Login button not working after password reset

Invoice export generates blank PDF

Steps to Reproduce

Steps to reproduce help developers recreate issues.

Each step should be clear and sequential.

Reproduction steps are one of the most important parts of a bug report.

1. Login to application
2. Navigate to Invoice Page
3. Click Export PDF
4. Observe blank PDF

Expected Result vs Actual Result

Expected Result describes how the application should behave.

Actual Result describes the issue observed during testing.

This comparison helps developers understand defects clearly.

Severity and Priority

Severity defines how serious the issue is.

Priority defines how urgently the issue should be fixed.

Critical production failures usually have high severity and priority.

Attaching Screenshots and Logs

Screenshots help visualize issues clearly.

Error logs and console messages improve debugging.

Attachments improve communication between testers and developers.

Bug Reporting Tools

Several tools are used for bug tracking and management.

Popular tools include Jira, Azure DevOps, Bugzilla, Trello, and Redmine.

Modern testing teams commonly use integrated project management systems.

Common Types of Bugs

Software projects commonly encounter different types of bugs.

1. Functional Bugs
2. UI Bugs
3. Validation Bugs
4. Performance Bugs
5. Security Bugs
6. Compatibility Bugs

API and Backend Bugs

API bugs affect backend services and integrations.

Examples include invalid responses, server errors, authentication failures, and incorrect data handling.

```
GET /api/students
```

```
Expected : 200 OK
```

```
Actual   : 500 Internal Server Error
```

UI and UX Bugs

UI bugs affect webpage appearance and responsiveness.

UX bugs affect usability and user experience.

Examples include alignment issues, broken layouts, and navigation problems.

Security and Performance Bugs

Security bugs may expose sensitive data or vulnerabilities.

Performance bugs affect application speed and scalability.

Security and performance testing are critical in enterprise applications.

Bug Reporting Best Practices

Write clear and concise bug reports.

Always verify issues before reporting.

Attach proper screenshots and logs.

Use meaningful titles and descriptions.

Maintain professionalism during communication.

Common Mistakes to Avoid

Avoid incomplete bug descriptions.

Do not report duplicate bugs unnecessarily.

Avoid emotional or aggressive communication.

Do not skip reproduction steps.

Sample Bug Reports

The following example shows a professional bug report structure.

```
Bug ID       : BUG-101
Module      : Login
Severity    : High
Priority     : Critical
```

Issue:

User unable to login after password reset.

Expected:

User should login successfully.

Actual:

Application displays invalid credentials error.

Summary

Bug reporting is an important part of software quality assurance.

Professional bug reports improve communication and debugging efficiency.

Clear reporting standards help development teams resolve issues faster.

Understanding bug reporting standards is important for testers, developers, and project teams.