

Software Defect Reports

2 key understandings:

- Firing errors, not just finding them, is the purpose of testing.
- Understanding the lifecycle and components of the defect report aids in communication between dev. and testing.

Purpose of Software Defect Reporting:

To ensure reliable and fast elimination of failures detected by the various test levels, a well-functioning procedure for communicating and managing those incident reports is needed.

↙(Software Testing Foundations)

Software Defect Report Synonyms:

SDR	Software Defect Report
SUR	Software User Report
PTR	Program Trouble Report
SCR	Software Change Request
DR	Defect Report

Software Defect Report Process:

- 1 Analyze the bug for reporting
- 2 Report.
- 3 Track the status of the fix & update
- 4 Retest the bug after fix is submitted
- 5 Close the bug after it is fixed.

1.1 Analysis of a software defect:

- > Find the root cause.
- > Determine if it is reproducible or repeatable.
- > Attempt to isolate the defect.
- > Investigate alternative paths to the issue.
- > Decide if it is worth reporting.

1.2 Analyzing software defect reports ensures proper prioritization:

- > Not all defects are created equal.
- > Determine the cause of the defect.
- > Additional information on how it can be isolated is valuable.
- > Report those defects which are important in the context

2.1 Reporting the software defect

- > Ensure it is not a duplicate.
- > Talk with the developer.
- > Enter it into the system.
- > Make sure it gets fixed.

2.2 Characteristics of an effective defect report -

- Numbered
- Understandable

- Simple
- Written
- Complete
- Explains the problem
- Includes min. no. of steps to reproduce.
- Be as concise as you can be.
- Provides a consistent means of communicating defects.

2.3 Components of a Software defect report →

1. Identifying information
2. Description of the problem
3. Various status indicators
4. Comments.
5. Miscellaneous information.
6. Supporting information.

2.3.1 Identifying information

- Unique number/ID
- Submitter
- Submit date.
- Program/product this is against.

- Version or revision of the product.

2.3.2 Description

- The problem itself

↳ What did happen

↳ What should have happened.

- The test case used.

- Any attachments / helpful info.

Also → > avoid vague or confusing terms (e.g. - frequently, sometimes)

> avoid uncommon abbreviations.

> use any standard terminology

> pay attention to grammar.

2.3.3 Status Indicator

- Overall report status

- Severity and priority.

how bad is it?

→ open/closed

↓
how urgent is it?

→ High -- critical defect; test cannot continue

Medium -- critical defect; test can continue with a workaround.

Low -- it's a problem but we can ignore it

Trivial -- documentation, help, files, mistakes

- Current resolution status.

Example of Resolution Status (Interim)

- None (this is what the board searches on)
- In process / assigned.
- Fixed ("pre-build" for an app defect)
- Ready for Test ("post-build", not used for defect in Test Suite)
- Retest
- Closed

(Final)

- Not repeatable
- Not a problem
- Do not fix.
- Duplicate

- Deferred.

2.3.4 Comments/Notes :

- Analysis Notes
- Resolution Notes
- Tester Notes
- Modification Notes
- Verification Notes
- Whatever-works-for-you Notes

2.3.5 Miscellaneous fields :

Steps to Reproduce

- Include setup information
- Anyone with the appropriate setup should be able to reproduce the problem
- Could be as simple as a Test Case ID, or as complex as a lengthy procedure
- Pay attention to problems that are merely repeatable and not reproducible

2.3.6 Supporting information:

- Excel print-outs.
- Flash drive w/ data/files

- Screen shots
- Error logs / trace files
- Test-case itself.



Logs are usually broader in scope & capture a wider range of events.

Traces provide more specific info. about the execution path of individual requests.



If a particular low-severity defect is such that by fixing it, it will cause many more problems to come up. Then DO NOT FIX!

3.1 Tracking a Defect

- Have a process
- Handled, primarily by the Defect Review Board.
- Include lead tester on the board.
- Ensure progress is made on defect.

4.1 Retesting the Defect

This can have 3 outcomes -

- Problem has indeed been fixed.
- Problem remains (unchanged)
- Problem is replaced by new one.

5.1 Close a Defect

- Add tester or verification notes.
- Make sure anyone else can also detect
- Close it (or recommend closure)

→ What if there is no "Defect Review Board"?

Tasks like...

- ... setting disposition status
- ... " progress status.
- ... " tracking the defects.

These tasks now fall to the testing team.
So,

- testers must take care at each step.

- anyone can enter a defect.
 - having a review board helps ensure quality of the process.
-

