

# Crash course on Reporting Bugs

---

## Santhosh Tuppad

Test Enthusiast

<http://tuppad.com> [Santhosh@testersblog.com](mailto:Santhosh@testersblog.com)

## Pradeep Soundararajan

Test Consultant, Coach, Author and Speaker

<http://testertested.blogspot.com> [Pradeep.srajan@gmail.com](mailto:Pradeep.srajan@gmail.com)

This project was initiated and first published in Utest. The authors hold the publishing rights to this article. This article is copyrighted under Creative Commons License . You are free to share this article with anyone by owing proper credits to the authors and without modifying any content.

One of the ways the world identifies good testers is based on their bug reports. A lot of credibility is at stake for a tester in bug reporting. We have witnessed, respect to a tester improving or tumbling down based on bug reports.

In this crash course, we share the practices of reporting bugs that were found to be of value to most of the stakeholders and customers we have served and are serving. Our intention is to help the testers' community, to learn a few things that are important to help boost their credibility, or re-iterate what they already know but probably don't practice. If you already practice these, wow!

## Audience of your bug report & Understanding their viewpoint

There are multiple audiences for a bug report. In this section we explore who the audiences are and what they want from a bug report.

**Programmers:** It is so obvious for most testers that their target audience is developers and yet they appear to pay little attention in understanding what the developers needs are. For those who worked as programmers and then turned to testers might understand what a developer might be looking for when reading a bug report but that doesn't necessarily mean developers turned testers are likely to report bugs in a way that caters programmers. It's a question of who has acquired and practices the skill.

A programmer might look for information

- To reproduce a bug
- To be able to understand the bug in a different view than reported
- To perform an investigation of what is causing it.
- To look for information about the bug and its co-relation to other bugs reported.
- To understand which part of code to touch to fix the bug.

**Co-testers:** This isn't obvious to some testers. Testers could be split across geography or across time. We often come across contexts like, a tester different from the one who reported the bug has to investigate the bug or test for the fix. So, a poorly written bug report could misguide a co-tester.

A co-tester might look for information

- To reproduce a bug in the same or different environment.
- To add more investigation notes to the bug.
- To provide additional information when it is deferred or rejected or even otherwise.
- To respond to a developers comment on the bug

**Test/Dev/Product Manager/Customers:** This segment of audience, are mostly decision makers or those who influence the decision makers. These people usually are in a situation where they don't have luxury in the world to read an entire bug report and then take a decision. Finding & reporting a bug is one part and making it useful to this audience is another.

A test/dev/product manager might look for information

- To take a decision of adding / not adding a specific module to the upcoming release
- To take a decision of ship/no ship of the entire product
- To understand how much more development / bug fixing work is needed.
- To plan a future release
- To help the customer plan releases to their customers

## Bug report elements & their significance

There are a lot of elements that constitute completeness of a bug report; here we discuss the most important ones and the ones that usually are erred quite frequently by quite a lot of testers.

**Summary:** The significance of summary is to get an idea of the bug as quickly as possible. For those who conduct bug triage, they might want to learn quickly if they should be picking up a bug or not for a specific meeting. A developer might want to learn if it pertains to the code he has written. A manager might want to know if the bug is in a specific module to which a release is being planned. Writing a concise yet meaningful

bug report is an important skill. How to improve bug summary writing skill? nt lyk dis! It requires work on English and understanding of audience, as a starter.

**Description:** This shouldn't be a copy paste of test case (in case you have it). It could contain information on what a tester consciously did and what was observed. To observe more, a tester might want to use focusing and defocusing heuristics while trying to repeat the actions performed that matter to run a specific test. Steps to reproduce aren't mandatory. We suggest that testers be context driven at least in this context. For a bug that is likely to be obvious to your audience, you might not want to say, "Step 1: Open the application. Step 2: Click on the menu and then see a Boom!"

Some organizations use bug reporting template that has limited flexibility in terms of the options and fields in the bug tracking system. However, a tester can report things relevant and important to a bug in the Description section. For instance, risk to the user, cost of not fixing the bug, how the bug could impact our business and other items related to bug advocacy can be provided within this section.

**Test Setup & Test Environment:** We do not have a section called "Test Setup" or "Test Environment" on most bug reporting templates and tracking systems but this is significantly important one, not just for you but for other audiences. Setup information can be provided as an additional note in the Description section. Information on Setup & Environment could involve configuring the system, making hardware level changes, disabling or enabling something relevant to the test or bug being reported. If a tester dumps a lot of unnecessary information in this section, it distracts the audience and makes the report less useful.

**Severity:** There must be at least half a million pages on the internet discussing about Severity and Priority. We don't wish to add to it but just like to mention "Be reasonable". Leave the priority to the business people and decision makers. Explain why something is of high/medium severity when it appears to be a candidate of "not so obvious" for your audience. Be open to learning of what others think about it.

---

## Drafting and Publishing Bug Reports

There are a couple of things a tester might want to consider doing when a bug is found, investigated and ready to start drafting a report.

### Looking for Duplicates

Why spend time on writing a bug report that is already written. If a tester finds a colleague or a fellow tester on the project has already reported the bug, then strategy changes from reporting to appending information on the already reported bug. This enhances the importance of the bug that is reported and ensures the management has lesser time dealing with duplicates during triage. This is a way of respecting the time of your audience.

## Use of auto spell and grammar check

Spell and Grammar check is extremely important to those testers whose English is of the second language. If you speak, write and read good English, no matter where you are born, English is one of those first languages for you.

Spell and Grammar checks are not fool proof. One of the examples we have encountered in a real time project is that of a report from a tester which reads like this, “X module feature is miss spelled”. You notice misspelled and miss spelled are meaningful words for the software although miss spelled means different thing to us from misspelled. *F u use sms lngage n rportin bgs ur hyks wil b shrnk d 2.*

## Use of screenshots

Screenshots help understand a bug faster. To help the audiences of a bug report, it is a good idea and already widely practiced to add screenshots for bugs that require it. Using JPG format is a wise idea. Some testers also point out what to look for and write short notes in the screenshot itself. When a tester needs a picture of a pencil, it is not wise to go to the moon and capture the whole earth to say, there is a pencil on this place.

## Use video for long steps

Some bugs are hard to describe or there could be language barriers between the test and dev teams. Usage of video recording of the bug is getting increasingly popular. The only danger associated with it is that – only few testers know for what bugs a video shall be helpful. It look obvious but it is not. When steps to reproduce are lengthy and hard to explain, usage of a video is wise. If the dev team is from a different country whose English is as bad as that of the test team (☺) then it is a wise idea to have a video of a bug that needs a lot of explanation. While videos could be of help to the developer audience, it could be a pain for others. So, a short description of the problem coupled with a video is the wisest choice we have made so far. *We don't suspect that you could be wiser than us.*

## Attaching document(s) of error logs

If the program logs actions and interactions of the system then it is likely integrated to be of help in bug fixing times. The log file could be huge and a tester who is context driven would copy paste only the relevant info in a text file and attach the same to the bug report.

## Usage of Compression Tools

Screenshots, Videos, Log files and other relevant files to a bug could eat up a lot of space and is a trouble for the audience to download each and every single file attached to a specific bug. Usage of compression tools, such as WinZip or 7Zip can help in such situations. *The worst tester would copy a single line of log; make a txt file and then zip the file. Well, it happened ☺*

## Naming convention for screenshot or video or documents

While naming image file or video file or document file, it is wise to follow a naming convention that goes well with audience. This helps in organizing and tracking these files easily. A worst tester would have a file crash.jpg and a good one would be likely to have something like Bug\_id\_Product\_module\_feature\_typeofproblem.jpg

## Checklist of publishing a bug report

- Look for duplicates
- Check for meaningfulness of entries in all Elements of the bug report.
- Try modeling as different audience and ask if the bug report is really useful to them.
- Make a list of mistakes in the past with bug reporting and run it through the report.
- Check for attachments, their sizes and relevancy to the context
- Save a copy of your bug report in MS Word or other editors as everyone knows bug tracking systems, crash, too.
- Finding bugs in the bug report and fix them

***“Those testers who write bad bug reports don’t become successful testers. Wouldn’t be completely wrong if we say when you are writing a bug report, you could be writing your own fate. That is why we hope you write it well.”***

---