

From <https://community.atlassian.com/t5/Marketplace-Apps-Integrations/How-to-write-a-defect-description/ba-p/1719660>

How to write a defect description?

By Katarzyna Kornaga, Atlassian

Testing is an integral part of software production and quality assurance. Bugs just happen, even within a top-notch programming job. To eliminate them and ensure the right quality of the app, each issue needs to be not only detected but also carefully described. How to report defects and how to write a perfect defect description?

What is a defect?

To put it simply, the defect is an error occurring in software that is under development, which makes it impossible to complete the intended action. Defects are usually results of programmers' slips, and they can appear at any stage of the software development process. This is why it's critical to start detecting them as early as possible, as the sooner they are found, the cheaper it is to fix them. Obviously, bugs can entirely ruin a project in terms of both technology and business, so the development team should bend over backward in order to eliminate as many of them as possible.

However, before we even start to fix errors, first we need to detect them - and describe them, too. This is just as important as testing itself, and doing it in the right way is just one of the tester's responsibilities.

How to report defects?

Accurate bug reporting is a crucial part of software testing. A bug report should contain particular information which allows assigning the issue to the responsible developer. Carefully prepared defect descriptions can help to avoid lots of misunderstandings, performing unnecessary jobs, or asking around for additional explanations.

Bugs are usually reported in dedicated bug management software. If you're a user of Jira, there are a number of plug-ins that you can check out in order to improve your defects reporting process. For instance, [you can check out Requirements & Test Management for Jira \(RTM\)](#) which includes ready-made layouts for efficient bugs description and shows relations between defects and other testing objects.

The screenshot displays the Jira RTM interface. On the left, a sidebar shows the project structure: 'Test project - J Software Project' with sub-items like 'UX board', 'Kanban board', 'Reports', 'Releases', 'Issues and filers', 'Pages', 'Components', 'RTM', and 'Project settings'. The main area is titled 'Defects' and shows a tree view of issues. The selected issue is '1.1.4. Manual control (id: 123)'. Below the tree, there are tabs for 'Details', 'Test Cases', 'Relations', and 'Tab'. The 'Details' tab is active, showing a '+ New direct link' button and a section for 'Direct relations and links'. This section explains that there are two types of relations: direct and indirect. It lists several 'Covers' and 'Included in' relations, each with a 'Summary of Issue' and a 'BACKLOG' status. At the bottom, there is a 'Link' button and a section for 'Indirect relations' which explains that indirect linking provides information about further relations between issues. Below this, there is an 'Executed by' section showing a 'FAIL' status for a 'Summary of Issue'.

Defects relations in RTM for Jira

How to report bugs? Before we proceed with more detailed explanations, let's start with some general defects-reporting advice:

- Before you report a bug, **verify whether anyone else has already done it**, especially if there are a few testers working on the project. Thanks to this, you will avoid ticket duplication and reporting issues which have been already fixed. In dedicated bug-tracking software, you will be able to easily monitor the status of each bug.
- **Make sure that the bug you want to report is actually a defect.** Try several times whether the bug is reproducible, check it also for the different configuration settings, different modules, and operational systems. If the situation keeps occurring in particular situations, it definitely needs to be reported.
- **Report bugs immediately.** There's no need to wait until the problem will escalate - each time you find the bug, describe it and report it

right away. If you decide to postpone writing a bug description, there is a chance that you will miss some important details.

How to write a good defect description?

As we already explained, writing a clear, consistent bug report is an integral part of software testing. A well-prepared bug description can accelerate the speed of defect elimination and make the entire software development process more effective. So, how to write a good bug report?

1. Assign a bug number

Each reported defect should have its unique ID number. It will enable identifying the issue by the responsible developer and fixing it as soon as possible. Remember that each bug should be reported separately and that their ID should not be confusing. If you're using a dedicated defect management system, such numbers will be generated automatically. In the case of Jira, the ID number is called a Jira Issue Key and is specific to every reported bug.

2. Prepare the right title

The title of a defect, called also a summary, is a crucial element of the description. It should immediately indicate the area of function in which it occurred and the type of the bug. Thanks to this, the responsible developer will be clearly informed about the defect and will be able to react immediately. Writing a consistent summary is important especially in more complex projects in which there are multiple errors that need to be split into different bug categories.

3. Be specific

Providing general information about the issue will definitely not be enough. To ensure efficient reporting and the fast reaction of the responsible developers, a tester should provide as much specific information as it's possible.

Thus, remember to name the exact product, version, component, platform, and operating system in which the bug occurred. Describe the situation in which the bug was revealed, so the reader can understand and visualize it. In order to write an even more complete bug description, you can also check error logs for more details. Such an approach will help to avoid misunderstandings and wasting time asking around for explanations.

4. Make it readable

Although the bug description has to include all the necessary details, no one expects essays. Try to describe the issue in a minimal number of words

and focus only on facts, not opinions. Every sentence of the report should be simple, meaningful, clear, and consistent - this will accelerate the defect-fixing process.

Moreover, remember to format your report. Use numeric lists and bullets, as they will make the report easily readable and clear-cut. Remember also to read it a few times before hitting the Submit button to make sure that everything will be obvious for the reader, and that your text will also be bug-free.

5. Indicate severity and priority

The next thing that needs to be done when preparing a defect report is rating its severity and priority. Although all bugs need to be fixed, not all of them are equally important. For instance, defects interrupting the crucial functionalities of the software have to be fixed immediately, unlike the small text or graphical corrections. Each bug should also be categorized, so developers can immediately know the area of expected improvements. Bugs prioritization and categorization will help to organize work more smoothly. In order to ensure the top efficiency, we highly recommend using dedicated software equipped with a bug scoring system, such as Jira extended with RTM and most popular but also configurable defect categories.

6. Write steps to reproduce

When preparing a defect report, write down step by step what exactly is happening. Avoid using general phrases, such as "I can't log in". Describe exactly what happens when the error occurs, and what steps are leading to it. Mind that there might be a few ways to complete one action, so the more specific you will be, the better. Define who, what, where, when and how you found the defect, so the developer could understand it and respond to the problem accurately.

7. Include expected results

A good bug description is not only about the problem itself. It should also include information about actual results vs expected results. Writing down what are the intended actions will help to identify the source of the problem and decide on what exactly needs to be fixed.

When writing about the expected result, remember to apply the general and well-known rule - be as specific as possible.

8. Provide documentation

If only it's possible, provide additional documentation of the detected bug. It can be a screenshot or a video in which the defect is visible. To explain

the issue better, you can even attach a few files presenting what happens when the defect occurs. In the picture, you can also highlight the most important remarks.

Such an approach will help the developer to understand the issue better. Of course, you can do it in a bug-reporting program that is equipped in a field in which you can attach additional files.

9. Avoid commanding tone

The last but not least, remember to be empathetic when writing a bug report. Never use harsh words. Mistakes happen and are an integral part of software development. So, don't blame your colleagues for the defects, even though it's them who messed up. Such attitude only leads to an unhealthy work atmosphere. It's not worth it! Instead of giving orders, try to write your report in a tone of request or suggestion. Believe us, this will sound much better.

The screenshot displays the Jira RTM interface for a defect titled "TP-1 Error message while trying to log in". The interface is organized into a sidebar on the left and a main content area on the right. The sidebar contains navigation options such as "Test project - J", "UX board", "Kanban board", "Reports", "Releases", "Issues and filters", "Pages", "Components", "RTM", and "Project settings". The main content area shows the defect details, including the title, status (BACKLOG), priority (Medium), assignee (John Doe), reporter (Jane Smith), fix version (1.1), components (Login), RTM environment (Chrome), and creation date (01/01/2020). The description of the defect states: "Each user should be able to log in after filling up matching 'username' and 'password' fields followed by clicking the 'log in' button. The bug is produced only in Chrome browser." The steps to reproduce are: "Open the app in Chrome browser", "Sign in with input: username: test, password: Test123", and "Click the log in button". The actual result is: "Under the fields there is an ERROR message, despite the fact that the data is correct." The expected result is: "The next screen with user's profile data is displayed." There is an attachment named "Login defect.jpg" and an activity section with a "Text field" and a "Pro tip: press M to comment".

Example of a well-described defect in RTM for Jira

If you'd like to make your bug-detecting process more efficient, we highly recommend using dedicated bug-reporting software. Such programs are equipped with all the functions automatically preparing clear-cut, consistent defect descriptions. By following the indicated steps, you can be sure that the format is right and that no crucial elements are missing.