**Bug Reporting**

Bug reporting in software testing is the process of documenting and communicating software issues to developers. It's a key part of software development and quality assurance, helping to ensure that the final product meets quality standards.

Developers are often under a ton of pressure to solve issues quickly without actually having a lot of time on their hands.

They usually face two extremes: *too much unhelpful information* or *too little important information*.

For every bug report, we highly recommend using a [**bug tracker**](https://marker.io/blog/bug-tracking-tools) like Trello, Jira, Asana, GitHub or GitLab and implementing a consistent, standardized approach.

Here’s how to write a bug report:

**1. Title/Bug ID**

Keep it short and specific.

Make sure it clearly summarizes what the bug is. Having a clear title on your bug report makes it easier for the developer to find later on and merge any duplicates.

Examples:

❌ Bad: *"I can't see the product when I add it, for some reason I try and it doesn't. WHY? Fix it ASAP."*

* vague
* aggressive
* too verbose
* asks for a solution to be implemented

✅ Good: *“CART - New items added to cart do not appear”*

* it helps developers instantly locate the issue (CART)
* it focuses on the actual technical problem

When developers review it, they’ll be able to instantly assess the issue and move on to other elements of the bug report.

**Visual proof/screenshot**

We all know that a picture is worth a thousand words. That stands true for bug reporting.

While it may not tell the whole story, a screenshot or video can add a lot of value by getting your developers to see and understand the problem faster.

**3. Expected vs. actual results**

When you report a bug, take some time to explain to your developer what you expected to happen and what actually happened.

Example:

Expected result: “Item should be added to the cart when I click ADD”

Actual result: “Item does not appear in the cart”

**4. Bug severity and priority**

By defining the severity or priority of the issue in your bug report, your developer understands how quickly a bug should be fixed.

The severity of your bug can be defined by the level of impact it has on your website or product. Once this has been determined you can label it as:

* Critical
* Major
* Minor
* Trivial
* Enhancement

The priority helps your developer determine which bug they should investigate and fix first. Here you can choose between:

* High
* Medium
* Low

As the bug reporter, you will normally be responsible for identifying the severity and priority.

Pro tip: it can be difficult, as the end-user, to determine bug priority/severity—but if it severely affects functionality and user experience, it’s critical!