BUG REPORTING: SUBMIT ISSUES LIKE A PRO

ARTUR GULA

NOT ONLY FOR SOFTWARE TESTERS

BUG REPORTING: SUBMIT ISSUES LIKE A PRO

NOT ONLY FOR SOFTWARE TESTERS

Artur Gula

Table of Contents

WHY SHOULD YOU READ ABOUT BUG REPORTING?

Let me quote some of my friends:

- 'I like fixing the bugs which you report,' they say.
- 'I'm glad that you test our software,' my colleagues constantly repeat.
- 'When you report a bug, I always know what to do!' I used to hear that many times.

Do you want to be that guy too? The person who provides precise but not overwhelming bug reports? I'll guide you step by step through the bug reporting process — no matter where you start. Even if you have never reported any bug, you will do it like a pro at the end.

IS THIS HANDBOOK FOR YOU?

I dedicate this handbook mainly to software testers or quality assurance engineers. But, it may be helpful for anyone who works with the software apps. Even developers report bugs — the same for business analysts, product owners, or project managers.
I firmly believe that you and your teammates will find many benefits after implementing the methods described in this manual.

FEW WORDS ABOUT ME

My name is **Artur Gula**. I started my journey with software testing in 2009. It all started from Ron Patton's classic book about software testing. Then I participated in more than 40 software projects. Currently, I'm a project manager and business analyst.

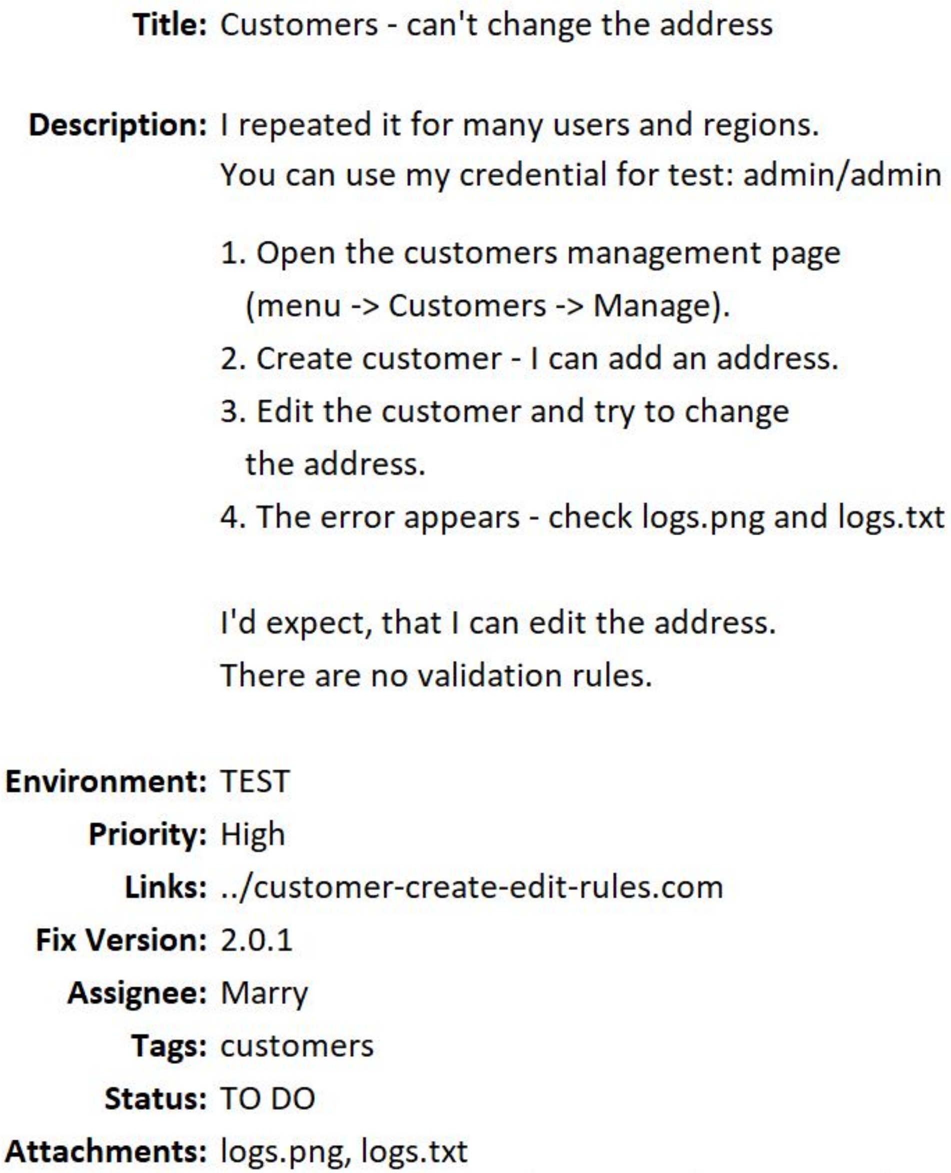
But I still enjoy testing software and reporting defects!

PERFECT BUG REPORT

Let's look at the exemplary bug report on the next page.

Our goal is to understand what to do in each section. I'll also point out what to avoid, so you will not fall into common traps.

Are you ready?



Title:	Customers - can't change the address
Description:	I repeated it for many users and regions. You can use my credential for test: admin/admin 1. Open the customers management page (menu -> Customers -> Manage). 2. Create customer - I can add an address. 3. Edit the customer and try to change the address. 4. The error appears - check logs.png and logs.txt I'd expect, that I can edit the address. There are no validation rules.
Environment:	TEST
Priority:	High
Links:	../customer-create-edit-rules.com
Fix Version:	2.0.1
Assignee:	Marry
Tags:	customers
Status:	TO DO
Attachments:	logs.png, logs.txt

Illustration 1: Exemplary bug report

0. BEFORE YOU START - 3 RULES

Before jumping into the details, I have three must-have rules for you.

Less means more

Your goal is not to write another episode of 'Harry Potter.' So the bug report must be clear and short. It must provide enough data for fixing it. And nothing more.

I recommend following the rule: separate bug reports for each defect. Do not combine many not-related issues in one ticket. It will be confusing hard to maintain and fix. Simpler means better!

And one more - some of your colleagues may not be native speakers. So do not try to be a poet. Instead, use uncomplicated phrases and essential words.

Facts not people

What do you feel when someone points out that you are wrong? It may be anger, offense, irritation. It's a similar mechanism when you report an issue in someone else's code. Your bug reports say: 'you made a mistake, men.' Of course, you can't change that. But the form of the issue may affect that feeling in both directions.

In the defect description, you should focus just on facts:

- What happened?
- When did it happen?
- Under which conditions?

Avoid judging the person or their work. Phrases like: 'it's stupid, it makes no sense, you are lazy, you don't understand, etc.' are unnecessarily aggressive. They don't bring any value and don't help fix the issue.

We don't have the right to judge anyone in private or professional life. So please don't do it, and the quality of your work and relations will improve.

Reproduce the bug

Do you know what would happen when you report a bug that you can't reproduce? 99% of developers will return it as 'unable to reproduce.' That remaining 1% is for newbies who don't know the rules yet. No reproduction means no fixing.

By reproducing, I mean achieving the same result at every attempt. So you can crash the system whenever you want. Isn't it lovely?

There are some exceptions when you should still report the bug without the reproduction steps. I'll share those tricks with you at the end. At this moment, I assume that you know how to repeat the incorrect behavior.

The good news is that it's possible to reproduce most software issues. I hardly remember bugs, which I couldn't repeat. So it's sometimes a matter of time and persistence, and you will find that faulty path.

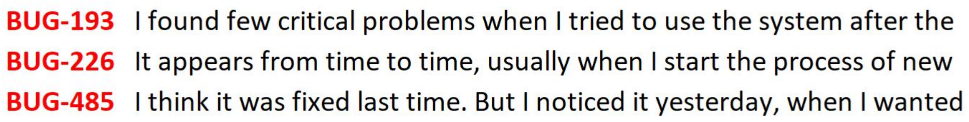
Did you find it? If yes, then let's report the bug!

I prioritized chapters based on their importance. If you don't have much time, just read a few sections from the top. It will impact your bug reporting process significantly.

1. TITLE

Please look at the list of tickets from the bug tracking system. Titles are visible up to some number of characters. It's a common constraint of the tools.

Now tell me, which one is about a login page?



BUG-193	I found few critical problems when I tried to use the system after the
BUG-226	It appears from time to time, usually when I start the process of new
BUG-485	I think it was fixed last time. But I noticed it yesterday, when I wanted

Illustration 2: Titles of bug reports

You don't know? Mee too!

Your goal

That example shows how important it is to use good titles for the bug reports. Programmers often scan the list looking for the bugs from their areas of specialization — the same for testers and other people.

Another reason is to support searching for the specific bug. It's common when someone asks, 'have we reported this issue already?' Then people try finding the matching bug report. It's much easier when your titles are correct. You can't effectively manage the work with messy titles in your ticketing system. The goal is to write a short and unique title that points to a bug occurrence.

What to do?

Some hints for good titles:

- Start with **crucial information.**
- Use **short sentences** - a maximum of 12 to 15 words should be enough.
- The not grammatical form may be better than correct, but a long one.
- Use **unique** titles.
- Use acronyms or jargon, but only when every team member will understand them.
- Optionally - use **tags or keywords** at the beginning, for example, in brackets, like [tag1][tag2].

What to avoid?

On the other hand, anti-patterns are:

- Describe everything about the issue in the title.
- Add any subjective comments or judgments.
- Repeat the same title every time.
- Make typos - people will search for bugs using their summary.

Examples

I'll show you some good and bad examples:

- 'When I tried to open a page with a registration form, it didn't respond' → change it to 'Registration form doesn't respond.' (essential information at the beginning).
- 'System doesn't work' → change to 'Can't open the main page' (to be more precise).
- 'Annual report doesn't generate data for last year when turning off all the filters' → change to 'Annual

report — no data for last year' (don't describe everything in the title).

- 'I don't like the new design of the landing page' → the title could be 'Landing page is not intuitive' (avoid judgments, they are not bugs).
- If you want to use tags, it may look like '[Users] [Create] problem with special characters.'

BUG REPORTING - HOW DO EXPERTS DO THAT?

Software development teams spend long hours clarifying, updating, and processing incomplete bug reports. It costs money and generates conflicts.

Move your bug reporting process to the next level with that practical handbook. I described 15 key areas of an ideal bug report. I gave you many hints and best practices from the over 40 real IT projects. I prepared the checklist, so you won't forget about anything.

Create bugs like a pro, starting from today!

ABOUT THE AUTHOR

Artur Gula - I am passionate about delivering high-quality software that responds to customers' and users' needs.
I was a software tester and test manager, so I know the pain of poor bug reports.
I am currently more focused on requirements analysis and leading IT projects.

Feel free to contact me at:
linkedin.com/in/arturgula