Reporting Bugs

Yegor Bugayenko

Lecture #2 out of 8
80 minutes

The slidedeck was presented by the author in this YouTube Video

Joel Spolsky

"Every good bug report needs exactly three things: steps to reproduce, what you expected to see, and what you saw instead."

— Joel Spolsky. Painless Bug Tracking. https://www.joelonsoftware.com/2000/11/08/painless-bug-tracking/, nov 2000. [Online; accessed 07-02-2024]

Nicolas Bettenburg

"Well-written bug reports are likely to get more attention among developers than poorly written ones... Steps to reproduce and stack traces are most useful in bug reports. The most severe problems encountered by developers are errors in steps to reproduce, incomplete information, and wrong observed behavior."

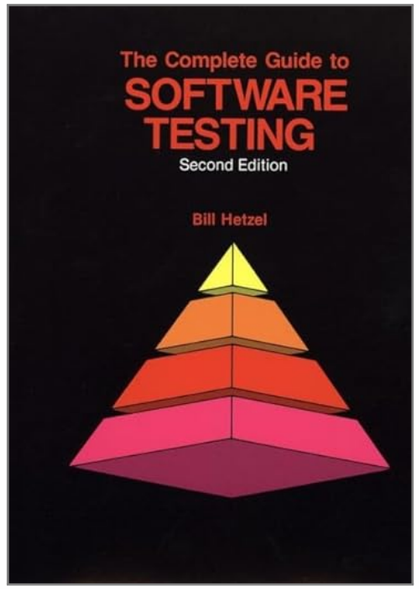
— Nicolas Bettenburg, Sascha Just, Adrian Schröter, Cathrin Weiss, Rahul Premraj, and Thomas Zimmermann. What Makes a Good Bug Report? In *Proceedings of the 16th International Symposium on Foundations of Software Engineering*, pages 308–318, 2008. doi:10.1145/1453101.1453146

Tommaso Dal Sasso

"The elements considered to be harder to provide [while reporting bugs] are the entity (e.g., class, file) that likely contains the defect, the steps to reproduce the failure, and a test case showing the defect."

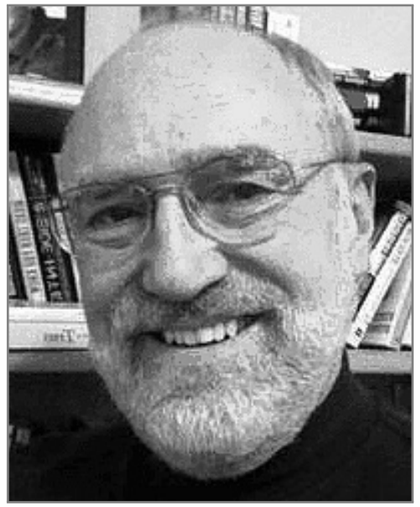
— Tommaso Dal Sasso, Andrea Mocci, and Michele Lanza. What Makes a Satisficing Bug Report? In *Proceedings of the International Conference on Software Quality, Reliability and Security (QRS)*, pages 164–174. IEEE, 2016. doi:10.1109/QRS.2016.28

1. Expect any program to have an unlimited number of bugs



"We can never be certain that a testing system is correct. These theoretical limit tells us that there will never be a way to be sure we have a perfect understanding of what a program is supposed to do (the expected or required results) and that any testing system we might construct will always have some possibility of failing. In short, we cannot achieve 100 percent confidence no matter how much time and energy we put into it!"

— Bill Hetzel. *The Complete Guide to Software Testing.* A Wiley-QED Publication, 1993



Myers, Glenford J.

"You cannot test a program to guarantee that it is error free... It is impractical, often impossible, to find all the errors in a program."

— Glenford J. Myers, Tom Badgett, Todd M. Thomas, and Corey Sandler. *The Art of Software Testing*. Wiley Online Library, 2004

2. Don’t ask questions or suggest features, report bugs instead

All of These Are Bugs:

- Lack of functionality
- Lack of tests
- Lack of documentation
- Suboptimal implementation
- Design inconsistency
- Naming is weird
- Unstable test

More about it by Bugayenko [2018a].

Reis Christian

“Though it commonly has a prejorative connotation, in the Mozilla Project the term **bug** is used to refer to any field request for modification in the software, be it an actual defect, an enhancement, or a change in functionality. (“bug-driven development”)”

— Christian Robottom Reis and Renata Pontin de Mattos Fortes. An Overview of the Software Engineering Process and Tools in the Mozilla Project, 2002

Kim Herzig

"In a manual examination of more than 7,000 issue reports from five open-source projects, we found 33.8% of all bug reports to be misclassified threatening bug prediction models, confusing bugs and features: On average, 39% of files marked as defective actually never had a bug."

— Kim Herzig, Sascha Just, and Andreas Zeller. It's Not a Bug, It's a Feature: How Misclassification Impacts Bug Prediction. In *Proceedings of the 35th International Conference on Software Engineering*, pages 392–401. IEEE, 2013. doi:10.1109/ICSE.2013.6606585

3. Reward yourself for each reported bug [Bugayenko, 2018c].

4. Report strictly one problem per ticket

John Anvik

"People play different roles as they interact with reports in a bug repository. The person who submits the report is the reporter or the submitter of the report. The triager is the person who decides if the report is meaningful and who assigns responsibility of the report to a developer. The one that resolves the report is the resolver. A person that contributes a fix for a bug is called a contributor."

— John Anvik, Lyndon Hiew, and Gail C. Murphy. Who Should Fix This Bug? In *Proceedings of the 28th International Conference on Software Engineering*, pages 361–370, 2006. doi:10.1145/1134285.1134336

5. Exaggerate for effect [Bugayenko, 2018b].

Do This, While Reporting Bugs:

- Stay cool
- Exaggerate
- Victimize yourself
- Push them
- Show efforts
- Look engaged
- Look altruistic
- Aggregate (not!)

More about it by Bugayenko [2018b].

6. Simplify until it's impossible to simplify any further [Bugayenko, 2022].

Bugs Occam's Razor

```
Here is my code:

a := 7
a := a + 5 - 3
a := a / 3
print a

It doesn't work as expected.
It prints 4, but it should
print 3.
```

```
Here is my code:

a := 7
a := a - 3
print a

It doesn't work as expected.
It prints 7, but it should
print 4.
```

Most probably, the subtracting operator doesn't do anything — this is the bug.

7. Be a prosecutor, not an advocate [Bugayenko, 2014].

> 8. Don’t give up until some changes are made to the code base [Bugayenko, 2014].

9. Prefer a disabled test in lieu of a bug report [Bugayenko, 2023].

Why not?

```
// @todo #42 This test is disabled
// because the fibo() doesn't work
// correctly with this input, returning
// 17711 instead of 28657. Fix it.
#[test]
#[ignore]
fn calculates_23rd_fibonacci_number() {
  let x = fibo(23);
  assert_eq!(28657, x);
}
```

"Such a PR serves as both a bug report (this is what the text of the puzzle will be turned into, once the PR is merged) and a test that reproduces the problem. It will be more than welcome by the repository maintenance team. This kind of PR saves the time they would spend creating a unit test. Also, it saves your time for creating a bug report, as it will be created automatically by the puzzles discovery tool." [Bugayenko, 2023]

Repositories With a Lot of Bugs (9 Feb 2024)

Github Repository	Issues
mozilla bugzilla	1,8M+
gitlab-org/gitlab	172,462
flutter/flutter	79,386
kubernetes/kubernetes	42,627
tensorflow/tensorflow	36,776
moby/moby	19,367

All repositories are open source.

References

John Anvik, Lyndon Hiew, and Gail C. Murphy. Who Should Fix This Bug? In *Proceedings of the 28th International Conference on Software Engineering*, pages 361–370, 2006. doi:10.1145/1134285.1134336.

Nicolas Bettenburg, Sascha Just, Adrian Schröter, Cathrin Weiss, Rahul Premraj, and Thomas Zimmermann. What Makes a Good Bug Report? In *Proceedings of the 16th International Symposium on Foundations of Software Engineering*, pages 308–318, 2008. doi:10.1145/1453101.1453146.

Yegor Bugayenko. Five Principles of Bug Tracking. `https://www.yegor256.com/141124.html`, nov 2014. [Online; accessed 07-02-2024].

Yegor Bugayenko. More Bugs, Please. `https://www.yegor256.com/180206.html`, feb 2018a. [Online; accessed 07-02-2024].

Yegor Bugayenko. The Right Way to Report a Bug. `https://www.yegor256.com/180424.html`, apr 2018b. [Online; accessed 08-02-2024].

Yegor Bugayenko. Either Bugs or Pull Requests... or You Are Out. `https://www.yegor256.com/180724.html`, jul 2018c. [Online; accessed 07-02-2024].

Yegor Bugayenko. Bugs Occam's Razor. `https://www.yegor256.com/220329.html`, mar 2022. [Online; accessed 07-02-2024].

Yegor Bugayenko. A Disabled Test in Lieu of a Bug Report. `https://www.yegor256.com/230725.html`, jul 2023. [Online; accessed 07-02-2024].

Tommaso Dal Sasso, Andrea Mocci, and Michele Lanza. What Makes a Satisficing Bug Report? In *Proceedings of the International Conference on Software Quality, Reliability and Security (QRS)*, pages 164–174. IEEE, 2016. doi:10.1109/QRS.2016.28.

Kim Herzig, Sascha Just, and Andreas Zeller. It's Not a Bug, It's a Feature: How Misclassification Impacts Bug Prediction. In *Proceedings of the 35th*

International Conference on Software Engineering, pages 392–401. IEEE, 2013. doi:10.1109/ICSE.2013.6606585.

Bill Hetzel. *The Complete Guide to Software Testing*. A Wiley-QED Publication, 1993.

Glenford J. Myers, Tom Badgett, Todd M. Thomas, and Corey Sandler. *The Art of Software Testing*. Wiley Online Library, 2004.

Christian Robottom Reis and Renata Pontin de Mattos Fortes. An Overview of the Software Engineering Process and Tools in the Mozilla Project, 2002.

Joel Spolsky. Painless Bug Tracking. `https://www.joelonsoftware.com/2000/11/08/painless-bug-tracking/`, nov 2000. [Online; accessed 07-02-2024].