

Assignment 1

EECS 4313 3.0 Software Engineering Testing, Winter 2015, Section M

Due: Monday, February 2, 2015, 2:00pm.

Format: In teams of two.

Editing and Evaluating bugs

The purpose of this assignment is to give you experience reporting bugs and developing test cases. This task will give you practice thinking how to develop a test case from a requirements document, what a professional bug report should be, as well as analyzing bugs before reporting them.

To get started

For any of the assignments in this course, you are free to work with any open source Java system you desire. The system must have a publicly accessible bug tracking system, so that you can submit bug reports, and the marker can access them. The course website includes links to several sites that host open source systems. The rest of the assignment refers to one such system that has worked well in the past. You are free to use it. If you choose a different system, you must adjust the steps below accordingly.

- Download the BORG Calendar installation package (link in course website), run it and follow the on-screen instructions. Remember your installation path defined in step 4. After product deployment, go to the installation path and run BORG Calendar by typing `./run_borg.sh` to get familiar with the application.
- Download the BORG Calendar product specification and explore the product website (both links on course website).
- Create a github account so you can submit bug reports (link in course website). The user name must start with your last name, and you must use your cse email account.

What to do

1. Submit 5 reports, either reporting bugs or proposing enhancements. For each report focus on the following:
 - Clarity and tone.
 - Attempt to replicate the bug.
 - Perform further analysis using various types of follow-up testing.

- Include important information that you successfully replicated the bug on XXX configuration in YYY build, or describing a simpler set of replication steps (if you found one). Your comments should NEVER appear critical or disrespectful of the developer.

Issues to consider when working on a bug report:

- Is the summary short (about 50-70 characters) and descriptive?
 - Can a developer understand the bug report? Is there sufficient detail to envision what the program did in response? Is it clear what the failure was?
 - Is it obvious where to start (what state to bring the program to) to replicate the bug?
 - Is it obvious what files to use (if any)? Is it obvious what you would type?
 - Is the replication sequence provided as a numbered set of steps, which tell developer exactly what to do and, when useful, what developer will see?
 - Does your bug report include unnecessary information, personal opinions or anecdotes that seem out of place?
 - Is the bug report too long? Too short? Does it have a lot of unnecessary steps?
 - Can you replicate the bug by following your steps?
 - Can developer get lost or wonder whether you had done a step correctly? Would additional feedback, e.g. "the program will respond like this...", have helped?
 - Does configuration or environment change have an effect on bug reproduction?
2. Create a written report that includes the report IDs or URLs for the 5 reports, an explanation of all reports, steps that you used to discover bugs and any follow-up testing you did to strengthen your bug report.

Since the marker may be unfamiliar with the functionality tested, you must explain what was the expected functionality and how you decided that your test case resulted in a failure. Include relevant screen shots if applicable.

Since you will probably NOT have time or the resources to run many follow-up tests, describe other tests which you would like to run. You should provide justification of these follow-up tests and explain what would you hope to learn from them.

What to Submit

Submit a PDF of your report electronically by typing the following:

`submit 4313 a1 a1.pdf`

where `a1.pdf` is your report. Your report must include the names and student numbers of both team members.