Association for Software Testing
Black Box Software Testing Series: Bug Advocacy Course

Assignment – Replicate and Edit Bugs

Cem Kaner

Version 9.2, January 2009

Overview

The purpose of this assignment is to give you experience editing bugs written by other people. This task will give you practice thinking about what a professional report should be, before you start entering your own reports into this public system.

The guidelines that I provide for evaluating bug reports are based on criteria developed by several test managers to help them evaluate the quality of their staff's work.

Phase 1

You'll start by submitting comments on one UNCONFIRMED report on the IMPRESS (or Presentation) product. This includes:

- comments designed to improve the report—make these as your changes to the report stored in the OpenOffice tracking system
- comments designed to evaluate the report—make these as comments on our forum

Phase 2

- Peer review the Phase 1 tasks (both sets of comments) from two other participants in the course.

Phase 3

In the third phase, you will pair up with another participant. Pick one UNCONFIRMED report on IMPRESS each. These must be different reports and they should be different from the reports that class has already worked on.

- before you submit any comments to the OpenOffice database or to our forum, send your comments to your partner and get his or her feedback first. Then improve your work as appropriate and after that,
- file comments designed to improve the report with the OpenOffice tracking system
- file comments designed to evaluate the report on our forum
- file comments on how the peer review helped (if it did) and whether you think it was worth it (and why)

Phase 4

In the fourth phase, you will grade two of the reports submitted in Phase 3, using the grading guidelines that we provide.

How to Evaluate a Bug Report

This collection of questions helps me quickly spot (and name) the weaknesses of bug reports. Don't ask every question about every report. When writing an evaluation, highlight those answers that seemed the most insight-producing.

Do ask **at least one** question within each of the four categories:

1. What are your **first impressions** of the report?
2. What happens when you attempt to **replicate the report**?
3. What **follow-up tests** should have been done in the course of writing this report?
4. Does the report include **speculation or evaluation** by the tester, and if so, is it appropriate and useful?

Skim through this list as you read the report—don't work through every question. Your evaluation should point out the strengths of what you have read as well as the weaknesses.

Evaluation: First impressions

- Is there a summary?
    - Is it short (about 50-70 characters) and descriptive?
- Can you understand the report?
    - As you read the description, do you understand what the reporter did?
    - Can you envision what the program did in response?
    - Do you understand 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 you exactly what to do and, when useful, what you will see?
- Does the report include unnecessary information, personal opinions or anecdotes that seem out of place?
- Is the tone of the report insulting? Are any words in the report potentially insulting?
- Does the report seem too long? Too short? Does it seem to have a lot of unnecessary steps? (This is your first impression—you might be mistaken. After all, you haven't replicated it yet. But does it LOOK like there's a lot of excess in the report?)
- Does the report seem overly general ("Insert a file and you will see" – what file? What kind of file? Is there an example, like "Insert a file like blah.foo or blah2.fee"?)

Evaluation: Replicate the Report

- Can you replicate the bug?
- Did you need additional information or steps?
- Did you have to guess about what to do next?
- Did you get lost or wonder whether you had done a step correctly? Would additional feedback (like, "the program will respond like this...") have helped?Did you have to change your configuration or environment in any way that wasn't specified in the report?
- Did some steps appear unnecessary? Were they unnecessary?
- Did the description accurately describe the failure?
- Did the summary accurate describe the failure?
- Does the description include non-factual information (such as the tester's guesses about the underlying fault) and if so, does this information seem credible and useful or not?

Evaluation: Follow-Up Tests

- Are there follow-up tests that you would run on this report if you had the time?
    - *In follow-up testing, we vary a test that yielded a less-than-spectacular failure. We vary the operation, data, or environment, asking whether the underlying fault can yield a more serious failure or a failure under a broader range of circumstances.*
    - *You will probably NOT have time to run many follow-up tests yourself. For evaluation, my question is not what the results of these tests were. Rather it is, what follow-up tests should have been run—and then, what tests were run?*
- What would you hope to learn from these tests?
- How important would these tests be?
- Are some tests so obviously likely to yield important information that you feel a competent reporter would have run them *and described the results?*
    - The report describes a corner case without apparently having checked non-extreme values.
    - Or the report relies on other specific values, with no indication about whether the program just fails on those or on anything in the same class (what is the class?)
    - Or the report is so general that you doubt that it is accurate ("Insert any file at this point" – *really? Any file? Any type of file? Any size? Did the tester supply* reasons for you to believe this generalization is credible? Or examples of files that actually yielded the failure?)

Evaluation: Tester's Speculation or Evaluation

Some bug reports include evaluative (rather than factual) remarks from the tester, such as hypotheses about the underlying fault or about the importance of the problem to the customer. The report need not include such information, but if it does, it should be credible, accurate, and useful.

- If the report includes such information, is it credible and useful or not?
- Does the report cite facts to support the evaluation?
- Is the evaluation based on special knowledge of the tester that might not be readily available to the programmer fixing the bug?

Background to the Assignment

We want you to develop experience in evaluating and editing bug reports for a few reasons:

1. We have been repeatedly shocked by the low communication quality of bug reports at good companies. That makes it important for us to work on your technical and persuasive writing skills as they apply to bug reporting.
    a. The simplest form of training--having students write bug reports and then critiquing the reports, doesn't always work well. With many writers, it takes report after report after report to get much improvement, as it does in many other courses that try to teach writing skills.
    b. A different approach to writing instruction, which is what we're trying here, is to teach you to be critical, analytical readers of *other people's* bug reports. As you see the common mistakes in these reports, identify them and explain what makes them problematic, you will develop better skills for evaluating your own reports.
2. It is common for companies to require testers to review bug reports before sending the reports to the programmers"
    a. This is especially common on open source projects, because the people who write the original bug report are often inexperienced observers and writers. Sending first-draft bug reports to volunteer programmers, often wasting their time, will drive away many volunteers.
    b. Even some of the more traditional organizations have testers review bugs (before they go onto the queue for programmers), especially design requests, bugs from new testers, and bugs from non-testers. Some independent test labs claim to review every bug before passing it to their clients.
    c. As you'll see, this process has benefits and costs. It takes a lot of time. If you are going into testing, or development management, having personal experience with the costs and the benefits will help you assess your bug reporting process.
3. As a manager, you will often be asked to evaluate your staff. This is very difficult. Feedback to staff on the quality of their work is often weak.
    a. Detail-oriented reading, that starts from a carefully considered list of questions about the quality of a given task, can give you a solid, fact-based foundation for staff training and decisions about raises, layoffs, and promotions.
    b. The evaluation framework that we use in this assignment is derived from materials actually used by test managers to review their staff's work.

The challenge in following this approach in this course is that many students are also weak critical evaluators of other people's writing. This skill is not widely taught to science and engineering students, so along with asking you to apply this style of evaluation once, we are using this assignment to give you practice with the approach.

Good testing requires development of several skills. The idea of this multi-phase assignment structure is to keep at the task, approaching it from a few different ways, until you get good at it. Our experience suggests that several students will need to work through all four phases and the feedback they get from them to finally cross over into doing obviously good work without struggling to achieve it.

Details of the Assignment

Phase 1

Submit comments on one UNCONFIRMED report on the IMPRESS (or Presentation) product. This includes:

1. **comments on the report stored in the OpenOffice tracking system, designed to improve the report**
2. **comments on our forum designed to evaluate the report**

Find a few bug reports in the tracking system that have not yet been independently verified. These are listed in the IssueTracker database as UNCONFIRMED.

- ***Don't feel that you have to make your comments on the first report you find. If the first one is too hard to work with, switch to another.***
- ***If you submit comments on reports marked NEW or something other than UNCONFIRMED will be rejected as nonresponsive to the assignment.***

Once you have chosen your report, you have two tasks:

1. ***EDIT the report TO IMPROVE IT*:** revise it in ways that help the OpenOffice developers prioritize the bug accurately and fix it efficiently. Be sure that your added comments add value, because every time you edit a bug report, IssueTracker sends an email to many members of the development team. Making a bunch of low-value edits to the bug reports is like making a denial-of-service attack on the time/attention of the developers. This can make them grumpy.

   Some common ways to improve the report include:

   - Add a comment indicating that you successfully replicated the bug on XXX configuration in YYY build.
   - Add a comment indicating that you were unable to replicate the bug, if that failure would be informative. For example, if troubleshooting so far indicates this is a Macintosh-only bug, successful replication on a Linux system would be interesting, but failure to replicate on a Linux system would be uninformative. If you are reporting failure, what variations of the reported replication steps did you try in your effort to replicate the bug.
   - Add a comment describing a simpler set of replication steps (if you have a simpler set). Make sure these are clear and accurate.
   - Add a comment describing why this bug would be important to customers. (This is only needed if the developers are likely to defer it because it seems minor or unimportant. It is only useful if you know what you are talking about).
   - Your comments should NEVER appear critical or disrespectful of the original report or of the person who wrote it. You are adding information, not criticizing what was there.

   If you edit the report in the database, **never change what the reporter has actually written**. You are not changing his work, you are adding comments to it at the end of the report

   Your comments should have your name and the comment date, usually at the start of the comment, for example: "(Cem Kaner, 12/14/09) Here is an alternative set of replication steps:")

2. ***EVALUATE the quality of the report*.**

   - Use some of the questions from the chart on How To Evaluate a Bug Report. Feel free to add some questions of your own. Try to ask and answer at least 10 questions.
   - Post your answers to our forum, NOT to the Open Office bug report.
   - This task should help you coach colleagues in bug report writing.

Phase 2

Phase 2 gives you the opportunity to see how other students handled the evaluation tasks. This will often lead to insights into both, what makes a bug report strong or weak, and what makes an evaluation strong or weak.

- **Please peer review the Phase 1 comments of two other participants in the course.**
    - **Evaluate their contribution to the bug report using the same 35-question framework as you used in Phase 1**
    - **Evaluate their evaluation of the bug report (the one they submitted on the forum).**

When I grade bug evaluations, I generally do the following:

- Find the student's bug in the database
- Replicate it
- Consider the student's comments within the bug report (the student was supposed to strengthen the report)
- Comment (to the student--in this course, on the forum), using the core criterion is whether the student's comments would help the dev team assess and troubleshoot the bug
- Review the student's comments to our class about the bug report
- The core criterion is whether the student's comments show insight into good communication and troubleshooting
- In making comments, tie them to the assignment itself, with its multi-page list of criteria under which we evaluate bug reports

In my experience, the process of evaluating other students' evaluations of bugs leads many of my students to new insights. This is an important part of the process.

Sample Evaluation and Peer Review

Part 1: Here is an original bug report from the Open Office database--***This is the input to Phase 1.***
Part 2: Following that, I include a student's evaluation of the report. ***This was an output from Phase 1.***
Part 3: Given the bug report and the student's evaluation, I wrote a Phase 2 peer evaluation.

Issue #:	***<number deleted>***	**Platform:**	PC	**Reporter:**	***<name deleted>***
Component:	Presentation	**OS:**	Windows 2000	**Add CC:**	
Subcomponent:	viewing	**Version:**	OOo 3.0 Beta		
Status:	UNCONFIRMED	**Priority:**	P3	**CC:**	None defined
Resolution:		**Issue type:**	DEFECT		
Assigned to:	cgu (cgu)				
QA Contact:	issues@graphics				

*** Summary:** try to run slide show, and impress crashes.

	Date/filename:	Description:	Submitted by:
Attachments:	Fri May 23 19:36:00 +0000 2008: test.ppt	ppt presentation created in 3.0 (application/vnd.ms-powerpoint)	***<name deleted>***
	Sat May 31 00:03:00 +0000 2008: trace.txt	Trace back from slide show crashing (text/plain)	***<name deleted>***

Description: Opened: Sat May 17 03:51:00 +0000 2008 Sort by: Oldest first | Newest first

Create slide show (3 slides!), try run slide show, and impress crashes. Happened three times. Sufficient processor speed/RAM on my laptop.

------- Additional comments from <OOo project lead> Mon May 19 07:22:25 +0000 2008 -------

Reassigned.

------- Additional comments from <name deleted> Tue May 20 08:54:14 +0000 2008 -------

Do you use empty slides?
If this occurs with a specific document please attach the document.

------- Additional comments from <name deleted> Fri May 23 19:36:45 +0000 2008 -------

Created an attachment (id=53892)
ppt presentation created in 3.0

------- Additional comments from <name deleted> Sat May 31 00:03:57 +0000 2008 -------

Created an attachment (id=54103)
Trace back from slide show crashing

------- Additional comments from <name deleted> Sat May 31 00:04:33 +0000 2008 -------

Could be a dup of this issue:
http://qa.openoffice.org/issues/show_bug.cgi?id=90085
attaching stack trace.

------- Additional comments from <BBST student> Sun Jun 1 03:03:39 +0000 2008 -------

This happens with the sample presentation at
http://www.sunvirtuallab.com:8001/sample_docs/109171/generic_fr.ppt
(winxp, beta3.0)
All other views are fine, but slide show crashes, though I did not get a

```
traceback to compare to the other issue.
```

==========

2. The student's (Phase 1) evaluation of this report

In a general sense, I agree with several other folks - you certainly see a broad spectrum of bug reports in their system. A lot of them don't provide a lot of detail - mostly because it seems that for the posters, the problem is persistent and easy to reproduce, so they assume it should be for everyone. This is especially true of folks reporting crash bugs.

http://www.openoffice.org/issues/sho...g.cgi?id=89572

I commented on this issue just to provide the information that it is easy to reproduce using the sample file provided in a different bug report I viewed - that sample includes a few languages, and sound.

Overall, this bug report was weak from the start - saying you have sufficient RAM/Memory is not a quantitative statement - just give the actual RAM/Memory, along with your exact OS information. Similar with the slides - there was no info given as to the features that might be on the 3 slides he's got; this particular issue may be related to similar stuff I read with issues on sounds.

Like someone else, I attempted to reproduce this issue also:
http://www.openoffice.org/issues/sho...g.cgi?id=89701

And even following the extra info the user provided, I had no issues at all with the steps he'd provided.

One thing that surprised me was more folks not including the file they're working with - a lot of times issues can be specific to some property or preference set in that file; not sure if they're encouraged not to, however, but it would seem to me that would make things far easier for the QA team and other folks attempting to reproduce their issues.

3. Phase 2 comments on the report

Phase 1 of the Assignment involved two subtasks:

1. Edit the report to improve it, and

2. Evaluate the quality of the report.

Phase 2 comments on "Edit the report to improve it"

Your edit does not state whether you were able to replicate the failure that was reported. In addition, it does not describe your system configuration. Because of this, I don't know how to tell whether your report is helpful or a distraction.

When I attempted to replicate this bug, I downloaded the bug reporter's file, was unable to replicate the crash, and noticed an oddity in the file (I can't move or stretch the graphic). I also downloaded the file you pointed to and did crash on that.

Note that the two files differ in an important respect: the first page of the one you point to has an animation.

My comment to the OOo team was:
------- Additional comments from phase2evaluator Sun Jun 1 19:03:10 +0000 2008 -------

```
I do not replicate this failure. I downloaded the attached file, ran the slide
show and it terminated normally. My system is WinXP, 4gB ram, dell m6300
notebook. I did notice with these slides that when I select the graphic on any
of the slides (I tried all 3), I get the selection boxes but cannot stretch or
reposition the graphic. Similarly, when I go to notes view, I can select the
slide (get the boxes at each corner) but I cannot stretch or move it.
I do replicate a crash with
http://www.sunvirtuallab.com:8001/sample_docs/109171/generic_fr.ppt and
therefore I think these may be distinct failures.
```

I think your comment was appropriate but would have been better if you had included

- your configuration information
- whether you could replicate the bug

Phase 2 comments on "Evaluate the quality of the report."

The assignment (pages 2-3) suggests several ideas for evaluating the bug. You cover some of them. No one would, or should, take the time to work through all of them for any particular report. That said, here is a slightly more structured set of comments on Issue 89572 (with a few notes on your comments)

First impressions:

- The summary is very broad: crash on slide show. OOo won't crash on EVERY slide show (or even most of them) and so I think there is critical information missing that I would expect in a summary. Still, it identifies severity and will show up in a search, so it meets several of the OOo project expectations.

- The replication conditions are very poor. The initial post implies that any 3-slide file will cause a crash. That's not true. The follow-up adds a 3-slide file (3 copies of the same slide) and a traceback. I still don't have any idea what is special about this file, if anything.

Replicate the Report

- I was unable to replicate the problem as reported, with the file supplied.
- The replication steps are, in essence, open my file and run a slide show, then you crash. This is insufficient.
- I don't know his configuration and so I don't know if this is a config difference. He is Win2k whereas I am WinXP.
- It's not clear why he creates a ppt file instead of an impress file. It's not clear whehter he created or imported the ppt file. His graphic is odd in that I cannot move or edit it, I don't know the properties of the graphic; could that be part of the problem? How did he create this file?

Follow-Up Tests

- I tried creating a file in Office (to make a ppt) with graphics, then importing into OOo. There was no failure.
- I tried creating a file in OOo (new file, insert a jpg graphic about the same size as the one in the reporter's test file, duplicate that slide twice to get three identical slides), saving it as ppt, exiting OOo, then opening the ppt into OOo, then doing a slide show. That all worked.
- I tried opening a long ppt file into OOo (the 197-slide bug advocacy slide set), in ppt and pptx formats. The pptx format was incorrectly read by OOo but did not cause a crash. The ppt file opened OK and I went through the full slide show w/o incident.
- I tried repositioning the graphics in the reporter's file and various other tasks to figure out what the graphic was (background? no...) without success.
- I also replicated the crash with the other file that you mentioned,

http://www.sunvirtuallab.com:8001/sample_docs/109171/generic_fr.ppt

so I can get a failure on this one, but not on the first one.

Overall evaluation of the Report

This was a weak report. It's style was OK but its information was insufficient.

Phase 3

In the third phase, you will pair up with another participant. Pick one UNCONFIRMED report on IMPRESS each. These must be different reports and they should be different from the reports that class has already worked on.

In this phase, before you submit any comments to the OpenOffice database or to our forum:

- **send your comments to your partner and get his or her feedback first. Then improve your work as appropriate and after that,**
- **file comments designed to improve the report with the OpenOffice tracking system**
- **file comments designed to evaluate the report on our forum**
- **file comments on how the peer review helped (if it did) and whether you think it was worth it (and why)**

At least two days before the start of Phase 3, you should tell the instructor who your partner is. Otherwise, the instructor will assign your partner.

The main mistake people make in Phase 3 is to ignore their partner, doing (and submitting) their work on their own. The second most common mistake is to give your work to your partner so late that s/he can't do anything effective with it.

Peer review makes a big difference in quality, but you won't experience this if you don't give it a fair chance.

If you're taking this as an online course, we recommend that you set up at least two phone calls to work through the details of your reviews of each other's work. Making four phone calls, one per day, would not be unusual. To make free phone calls across continents, we recommend skype.

Submissions should be just like Phase 1

- useful comments added the original bug report
- an evaluation report to the forum

but better.

Our **Phase 3** goal is for you to do a better Phase 1:

- better contribution to the OOo bug report (submitted to the OOo database)
- better evaluation of the OOo bug report (submitted to us)

The Phase 3 tools to support that are:

- peer review (private, between partners)
- repeated emphasis on the 35 questions in the framework. If you are curious about the instructional history of student-guiding frameworks (like this one), read up on the concept of scaffolding.

Coming out of Phase 3, we hope you see a big improvement in your work and your partner's as a result of the peer review. This is the best way we know to demonstrate the value of having testers review each other's bugs.

Phase 4

In the fourth phase, you will grade two of the reports submitted in Phase 3, using the grading guidelines that we provide.

The way we'll usually work this is to assign a report to you and one to your partner.

- You are responsible for both, but you will be the lead reviewer for the one submitted to you and the assistant reviewer for the other one.
- Both of you should read both reports, compare notes on them, and then you write one evaluation, your partner writes the other.

Your ultimate grade for this assignment will be subjective and holistic. That is, I suggest that you form an overall impression and base your grade on that.

The two tables below can help you develop your impression. Your instructor will post a copy of the tables online, so that you easily copy the tables into your evaluation posting and write your notes into them.

- Our hope for **Phase 4** is that when you evaluate the output from Phase 3, you will see work products that are a lot better than you saw in Phase 1.
    - If the work products really are good, this evaluation will be very fast.
    - If the work products are not so good, they are probably still better than Phase 1, but this gives one last training for improving future bug reporting. For several students, this last review is when they finally "get it."

0. THE TASK--HERE'S WHAT HAS BEEN SUBMITTED TO YOU

- A bug report number, that points you to a bug in the OOo database
- A bug report in the OOo database that has the comments of one student (or a partnered pair of students, but I'll pretend it is just one student)
- An evaluation report that tells us what the student thinks of the report.

Your total grade will combine your assessment of the bug itself (how the student improvement) and your assessment of the student's insight into the bug reporting communication, as reflected in the evaluation report.

Note that in the tables below, the score can reach a total above 10. That's because there are many ways to do a good job. I don't rely mechanically on the total. For example, suppose that in the evaluation report, the student does a broad but mediocre job (mediocre in terms of the quality of comments). The point total might be a 9 or 10 (of the total possible 15 points), but you might feel that it is C or B level work. In a holistic evaluation, you use categories to organize your thinking but ultimately follow your summary impression, not the score. So you would assign a grade of C or B to the work, rather than A.

PHASE 4 TASK 1. ASSESSMENT OF THE BUG ITSELF

Allow 10 points for the bug report itself. The critical criterion is the extent to which the student's comments on the bug should be helpful to the OOo team:

- For example, if the student is reporting a failure to replicate, what makes the failure persuasive/useful? Seems to me that there are at least 4 items:
    - clear statement that it is irreproducible
    - clear statement of the steps followed, especially if the original report is unclear. If the original is clear and precise, it is sufficient to say that the listed steps were followed
    - clear description of the variations attempted, that go beyond the steps.
    - clear description of the configuration tested. More configs are better.

- For example, if the student is responding to a design issue, what information is brought to bear to assess the value of the requested design change?
    - oracle used?
    - market data / books / documentation / whatever?
    - scenarios?

SCORE YOU ASSIGN AND YOUR NOTES	Comments on the bug in the Issue Tracker record	Points possible (& notes) (Max total is 10)
	Student's comments are disrespectful in tone: 0 on the assignment	Zero on the assignment
	A follow-up test or discussion suggests that the student doesn't understand the bug (doesn't understand what is being complained about, and that lack of understanding is the fault of the student and not the bug report)	Best possible grade on the report = C (6/10)
	Clearly reports a simpler or clearer set of replication steps	up to +5
	Clearly reports one or more good follow-up tests	up to +5
	Good argument / data regarding importance	up to +8
	Student's comments include configuration & build	+1
	If the bug is in fact not reproducible, provides good description of effort to reproduce the bug, credibly demonstrates that this bug is not present on this configuration	up to +5
	Report a failure to repro on an alternate configuration without noting the config difference	-1
	Report a failure to repro on an alternate configuration that was already dismissed as irrelevant (or known not to yield the bug)	-5
	Report a criticism / failure to repro / other info that was already reported in another comment (unless the redundancy is intended to add value). The key thing here is to not waste time of the programming team.	up to -5 for each redundancy
	OVERALL EVALUATION--MAKE A HOLISTIC JUDGMENT	A+ = 10, A=8, B=7,C=6, D=5

PHASE 4 TASK 2. ASSESSMENT OF THE STUDENT'S EVALUATION

The first four sections of the grading chart (below) are from How to Evaluate a Bug Report, on pages 2-3 of this assignment:

- First impressions
- Replication
- Follow-up tests
- Student's speculation or evaluation

The other questions ask for your subjective impressions:

- ***How well did the student do?***
- ***How much insight did s/he show?***

SCORE YOU ASSIGN AND NOTES	CATEGORIES IN THE STUDENT EVALUATION	POINTS POSSIBLE (& notes) (Max total is 10)
	First impressions	up to 3
	Replication	up to 3
	Follow-up tests	up to 3
	Student's speculation or evaluation	up to 3
	Your closing impression of the sophistication of the student's work	up to 3
	Your closing impression of the insightfulness of the student's work	
	Any additional notes on what was done well	
	Any additional notes on what was done poorly	
	Any additional notes on what was missing that should have been there	
	OVERALL EVALUATION--MAKE A HOLISTIC JUDGMENT	A+ = 10, A=8, B=7,C=6, D=5