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

Assignment – Replicate and Edit Bugs

Cem Kaner

Version 11.3, SUMMER 2011

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.

Phase 1

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

- If you can't find any good bugs, remember that not all bugs will be classed as "Defect." Think back to the lecture. What else counts as a bug?

- As soon as you find a bug to work on, post its bug number to the forum so that no one else works on the same bug

- File comments designed to improve the report as changes to the report stored in the OpenOffice tracking system. Use Exhibits 1 and 2 as a guide.

- Send comments to us that evaluate the communication quality of the original bug report. Make these as comments on our forum. Use Exhibit 2 as an example.

Phase 2

- Write two peer reviews. Evaluate the Phase 1 tasks (both sets of comments) from two other participants in the course, using the grading guidelines that we provide. *Note: you are not evaluating the original bug reports, you are evaluating how well your fellow student worked with those bug reports in Phase 1.*

- To guide your peer reviews, use the grading guidelines that we provide in Exhibits 3 and 4.

Phase 3

Phase 3 is the same as Phase 1 except that you will pair up with another participant. Pick one UNCONFIRMED report on IMPRESS each. These must be different reports. They should be different from reports that you or other members of the class have already worked on.

- As soon as you find a bug to work on, post its bug number to the forum so that no one else works on the same bug

- 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. Improve your work as appropriate. After you have your partner's comments,

  o file comments designed to improve the report with the OpenOffice tracking system

  o file comments designed to evaluate the report on our forum (use exhibits 1 and 2 as guides.)

  o file comments on how the peer review helped (if it did) and whether you think it was worth it (and why)

- *Sending your comments to your partner at the last minute is <u>unfair</u>*

Phase 4

Grade one of the reports submitted in Phase 3, using the grading guidelines that we provide in Exhibits 3 and 4.

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. Our approach is to help you learn 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 wastes their time and will drive away volunteers.

    b. Some traditional organizations have testers review bug reports 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 report 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 difficult and engineering managers tend not to be very good at providing detailed qualitative feedback to their staff.

    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.

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 many 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, to improve the report

    2. comments on our forum designed to evaluate the report

Use Exhibits 1 and 2 as guides and models for your work.

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, or if you submit comments on a product other than OpenOffice Impress, your work 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. 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. Wasting their time makes them grumpy and that makes us (your instructors) 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 might be uninformative. If you are reporting failure, be explicit about what variations of the reported replication steps you tried in your effort to replicate the bug.

   - If you can't replicate the bug on the newest version of OOo, but the original report was about a previous version, load that previous version on your computer and retest with that. If you still can't replicate the bug, report your efforts on both versions.

   - 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. Be careful about this. Don't make comments like this if you don't know what you're talking about. Don't make comments like this if te developers are likely to fix the bug anyway. This is a useful comment when the developers seem ready to defer the bug because they think it is unrealistic, and you know why it actually is realistic.

   - 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 questions from the chart on How To Evaluate a Bug Report (EXHIBIT 1). 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. Our goal is to lead you into insights into

- what makes a bug report strong or weak, and
- what makes an evaluation strong or weak.

Your task is to 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 (Exhibit 1)**
- **Evaluate their evaluation of the bug report (the one they submitted to us on our forum).**
- **Use the grading notes in Exhibits 3 and 4 to structure your evaluations.**

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

- Find the student's bug in the database
- Replicate it so that I can understand the student's comments. I am not interested in evaluating the original bug report. I am not planning to comment on the quality of communication of the original report. I am planning only to comment on what our student wrote as an evaluation of that report.
- Evaluate the student's comments on the bug report (the comments the student was supposed to strengthen the report). The core criterion is whether the student's comments would help the dev team assess and troubleshoot the bug. My comments are on our forum, not on the bug report itself. I don't expect to write any comments on the bug report itself.
- Review the student's comments to our class (on our forum) 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 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.

Some students refuse to do a genuine peer review. Instead, they just:

- repeat whatever was said by another student, or
- say something like "great work."

This doesn't help anyone. It doesn't give the student whose work you're reviewing any feedback. It doesn't teach you anything. All it does is waste the time of the people who try to grade your report.

Please, if you're not going to do the task, don't pretend to do it. Just don't do it at all.

We see the same kind of problem, and so we make the same request, for Phase 4.

Phase 3

Phase 3 is the same as Phase 1 except that you will pair up with another participant. Pick one UNCONFIRMED report on IMPRESS each. These must be different reports. They should be different from reports that you or other members of the class have already worked on.

As soon as you identify a bug that you are working on, post the bug number to our forum so that no one else will work on the same bug.

Before you submit any further comments to the OpenOffice database or to our forum,

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

Sending your comments to your partner at the last minute is unfair

As with Phase 1, use Exhibits 1 and 2 to guide your work.

If the instructor has not assigned partners, then 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:

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

The Phase 3 tools to support that are:

- o peer review (private, between partners)
- o repeated emphasis on the 35 questions in the framework (Exhibit 1). 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

Phase 4 is like Phase 2 but a little more formal.

You will grade one report submitted in Phase 3 using the grading guidelines provided in Exhibits 3 and 4. Please pay attention to the work this student submitted in Phases 1 and 2 so that you can comment on the presence or absence of improvement in this student's work over the three phases.

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 (Exhibits 3 and 4) can help you develop your impression. Copy these 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."
- We expect your Phase 4 evaluation to be more thorough and more sophisticated than your Phase 2 proposal.

0. THE TASK--HERE'S WHAT HAS BEEN SUBMITTED TO YOU VIA THE PHASE 3 FORUM

- 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 student's contribution to the bug itself (how the student improved it) and of
- the student's insight into the bug reporting communication, as reflected in the evaluation report.
- You will also comment on the presence or absence of improvement in this student's work over the three phases.

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

USE EXHIBIT 3

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

- o clear description of the variations attempted, that go beyond the steps.
        - o 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?
        - o oracle used?
        - o market data / books / documentation / whatever?
        - o scenarios?

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

USE EXHIBIT 4

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?***

THE EXHIBITS

FOR THIS ASSIGNMENT

- Exhibit 1: How To Evaluate A Bug Report
- Exhibit 2: Sample Of Bug Evaluation From A Previous Class
- Exhibit 3: A Grading Guide For Assessing The Comments That The Student Added To The Bug Report In The Issue Tracker Database
- Exhibit 4: A Grading Guide For Assessing The Comments That The Student Posted To Our Discussion Forum, That Evaluated The Bug Report In The Issue Tracker Database

EXHIBIT 1: HOW TO EVALUATE A BUG REPORT

<div>

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"?)

</div>

EXHIBIT 1 continued: HOW TO EVALUATE A BUG REPORT

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?

EXHIBIT 2: SAMPLE OF BUG EVALUATION FROM A PREVIOUS CLASS

Part A: This sample is based on an original bug report from the Open Office database--This is the input to Phase 1.

Part B: This is a slightly modified commentary from one of our students on the bug, in the Open Office database. This is the first part of Phase 1.

Part C: This is a slightly-modified student's evaluation of the report. This was an output from Phase 1

Part A: Sample Evaluation (PHASE 1)

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 <an OOo team member> 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 <original reporter of the bug> Fri May 23 19:36:45 +0000 2008 -------
```
Created an attachment (id=53892)
ppt presentation created in 3.0
```

Part B: Sample Evaluation (PHASE 1)

Our student added these comments

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

Created an attachment (id=54103)

Trace back from slide show crashing

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

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

Another student (probably the student doing the Phase 2 review) added these comments.

------- Additional comments from <another 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.

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

You certainly see a broad spectrum of bug reports in the OpenOffice 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.

EXHIBIT 2 continued: SAMPLE OF BUG EVALUATION FROM A PREVIOUS CLASS

The previous comments don't map well to the list of ideas in Exhibit 1. This assignment is from a time when we emphasized Exhibit 1 less than we do now. For current work, this would be a serious weakness in this evaluation.

Working from the Exhibit 1 structure, here's how I might have evaluated this bug report

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.

EXHIBIT 3: A GRADING GUIDE FOR ASSESSING COMMENTS THE STUDENT ADDED TO THE BUG REPORT IN THE ISSUE TRACKER DATABASE

SCORE YOU ASSIGN AND YOUR NOTES	What is your impression of the comments the student added to the bug reports in the Open Office database?	Points possible (& notes) (Max total is 10)
	Were the student's comments disrespectful in tone?	If disrespectful, zero the assignment
	Does the student's work (a follow-up test or discussion) suggest that s/he doesn't understand the bug report?	Best possible grade on the report = D (6/10)
	Does the student report a simpler or clearer set of replication steps?	up to +5
	Does the student clearly report one or more good follow-up tests?	up to +5
	Does the student provide a good argument or data regarding importance of the bug?	up to +8
	Do the student's comments include configuration & build information?	+1
	If the bug is in fact not reproducible, does the student provide a good description of effort to reproduce the bug? Does s/he credibly demonstrate that this bug is not present on her or his configuration?	up to +5
	Does the student report a failure to reproduce the bug on an *alternate* configuration without noting the configuration difference?	-2
	Does the student report a failure to repro on an alternate configuration that was already dismissed as irrelevant (or known not to yield the bug)?	-5
	If the student speculated about the cause of the underlying problem, was it clear that s/he knew what s/he was talking about?	If so, up to +3. If not, -3
	If the student discussed market impact or customer impact, was it clear that s/he knew what s/he was talking about? Would the comments have been useful and informative (rather than boringly obvious) to the dev team?	If so, up to +3. If not, -3
	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 ABOUT THE VALUE OF THE STUDENT'S COMMENTS FOR THE DEV TEAM WORKING WITH THIS BUG REPORT.	A+ = 10, A=8, B=7,C=6, D=5

EXHIBIT 4: A GRADING GUIDE FOR ASSESSING COMMENTS THE STUDENT POSTED TO OUR DISCUSSION FORUM, THAT EVALUATED THE BUG REPORT IN THE ISSUE TRACKER DATABASE

SCORE YOU ASSIGN AND NOTES	The student sent us a report on the quality of the report filed by the original bug reporter. Even if the original bug reporter was "a student", when we say "the student", we mean the person whose work you are evaluating and not the original author of the bug report. Record here your evaluation of the student's evaluation of the bug reporter's work.	POINTS POSSIBLE (& notes) (Max total is 10)
	How well did the student raise issues from the *First Impressions* part of Exhibit 1 in her or his evaluation?	up to 3
	How well did the student raise issues from the *Replication* part of Exhibit 1 in her or his evaluation?	up to 3
	How well did the student raise issues from the *Follow-Up Tests* part of Exhibit 1 in her or his evaluation?	up to 3
	How well did the student raise issues from the *Tester's Speculation and Evaluation* part of Exhibit 1 in her or his evaluation? Note: you are not considering (here) the student's own speculation about the cause of the bug or the market impact of the bug. You are asking whether the student considered the speculation/evaluation by the original reporter, and if so, how thoughtfully.	up to 3
	If the student did discuss the underlying cause of the bug or the customer impact, in her or his report to us, does s/he know what s/he is talking about? Do her comments give us more insight into the quality of the original report?	up to 3
	Your closing impression of the sophistication of the student's work	up to 3
	What is your overall impression of the insightfulness of the student's work?	-5 to + 5
	Do you have any additional notes on what the student did well in her or his evaluation of the original report?	-5 to + 5
	Do you have any additional notes on what the student did poorly in her or his evaluation of the original report?	-5 to + 5
	Any additional notes on what was missing that should have been there?	-2 to +2
	OVERALL EVALUATION--MAKE A HOLISTIC JUDGMENT	A+ = 10, A=8, B=7,C=6, D=5