The Quality Mark

The Newsletter for the Kitchener-Waterloo Software Quality Association

Winter 2001 Volume 1, Issue 2 http://www.kwsqa.org/

Ode to a Bug

By Paul Carvalho

Don't worry; I don't plan to break into rhyme (this time). The theme for this issue is, however, all about bugs. Okay, not *all* about them, otherwise you would likely be holding a 40-page newsletter!

So what is a bug? One definition might state that a bug is 'a flaw in design, coding or manufacture of software, which causes all – or some portion – of a program to not perform as expected.' There are other definitions. In fact, just about every book and resource you can find will have some definition for bug or defect. Bugs come in various shapes and sizes, and have many different names, including: defects, issues, software change requests, typos, product problems, enhancements, inconsistencies, errors and crashes.

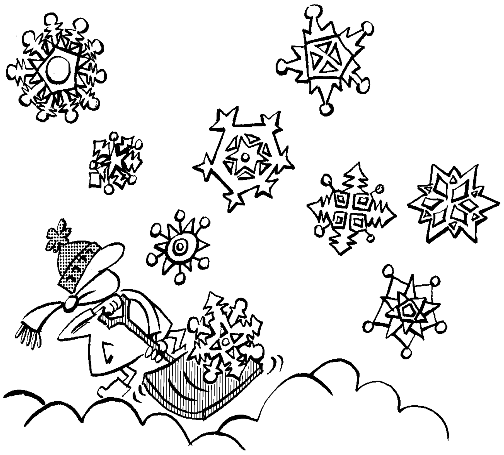
So what do we do with these bugs? Well, as professional bug hunters, we are responsible for observing, classifying and reporting bugs as we find them.

Observing: Finding bugs are, unfortunately, usually quite easy. There are many different techniques and approaches that one can employ – from simply "winging it," to a more structured approach using test case design methodologies and techniques. Sometimes you set the traps to see what you can catch, while other times you simply open the door (or window) and just start stepping all over them.

Classifying: Anyone worth their weight in salt will have some system in place to help classify the bugs found. If you don't already have one, get one as soon as possible! You ***must*** be able to *objectively* determine the severity of the defects found (e.g. Critical, Serious or Minor.) Classification often includes categories such as: Data integrity/loss, Hardware, Documentation, Design, GUI/Cosmetic, Suggestions and others.

Reporting: How well you report a bug directly affects how likely the programmer is to fix it. Sometimes, especially when you are working in a rapid development environment where the specifications (if any) are in flux, you need to be persuasive in your reports in order to get bugs fixed. Remember that writing is a skill. Speak to your Team Lead and/or the Technical Writers in your company for help in developing your ability to write clearly, concisely, and effectively if you think you need it.

Of course, not every bug found and reported will get fixed, but we must be ready to champion to get all the important bugs fixed. Grab your butterfly nets and notepads, keep your chin up, play fair and have a great time!



Inside This Issue...

When you are a Bear of Very Little Brain, and you Think of Things, you find sometimes that a Thing which seemed very Thingish inside you is quite different when it gets out into the open and has other people looking at it.

-- A.A. Milne, from The House at Pooh Corner

Quality Article

"Classic Testing Mistakes" - by Brian Marick

A Review by Terri Zuccherato

It's easy to make mistakes when testing software or planning a test effort. Some mistakes are made so often, that they deserve the label "*classic mistake*" as found in Brian Marick's article by the same name.

1 **The Role of Testing:** The testing team is *not* responsible for assuring quality. The testing team cannot improve quality, only enforce a minimum level.

Two items are often overlooked when testing which causes mistakes within the testing role:

- Too many bug reports from testers are minor or irrelevant, and too many important bugs are missed. That poses 2 important questions: (1) What is an important bug? And then, (2) Important to whom? To a first approximation, the answer must be "to customers."
- If usability problems are not considered valid bugs, your project defines the testing task too narrowly. Testers are restricted to checking whether the project does what it is intended, not whether what was intended is useful.

☞ **Helpful Hint:**

Tests designed before coding can improve quality. The process of designing them can find user interface and usability problems before expensive rework is required.

2 **Planning the Testing Effort:** To find the important bugs that customers see you need to write tests that will cross functional areas by mimicking typical user tasks. This type of testing is called scenario testing, task-based, or use-case testing.

Two related classic mistakes are:

- Not testing the documentation
- Not testing installation procedures.

☞ **Helpful Hints:**

- It is better to know something about all areas than everything about a few.
- Ask everyone you can for his or her opinion. Gather data from developers, marketers, technical writers, and customer support people.
- Use historical data. Analysing bug reports from past products will help tell you what areas to explore in this project.

3 **Personnel Issues:** Good testers require deep knowledge and experience. There are two classic mistakes when it comes to staffing a test team:

- Using testing as a transitional job for new programmers
- Recruiting testers from the ranks of failed programmers

continued on page 3

"Quotable Quotes"

We are what we repeatedly do. Excellence, then, is not an act but a habit.

-- Aristotle

Any activity becomes creative when the doer cares about doing it right, or doing it better.

-- John Updike

The act of designing tests is one of the most effective error prevention mechanisms known. The thought process that must take place to create useful tests can discover and eliminate problems at every stage of development.

-- Boris Beizer

He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.

-- Chinese proverb

I love deadlines. I like the whooshing sound they make when they fly by.

-- Douglas Adams

☞ **Helpful Hint:**

Hire some testers that are domain experts, some with programming skills, and seek out applicants from the customer service staff or technical writing staff. Create a testing team that has diversity. All of the members will lack some skills, but the team as a whole will have them all.

4 **The Tester at Work:** A tester who is not systematic, who does not spend time laying out the possibilities in advance, will overlook special cases.

Poor bug reporting is a classic mistake. Tester bug reports suffer from 5 major problems:

- They do not describe how to reproduce the bug
- They don't explain what went wrong
- They are not persuasive about the priority of the bug (explain why the customer would view it the way you do)
- They do not help the programmer in debugging
- They are insulting, so they poison the relationship between developers and testers

☞ **Helpful Hints:**

- A test case should contain a description of the setup, inputs given to the product, and a description of expected results.
- Testers can benefit from a second pair of eyes. Good testers are masters at noticing "something funny" and acting on it. Good testing is a combination of following a script and using it as a jumping-off point for exploration of the product. It is important to check that the product does what it's supposed to do, *and* that it doesn't do what it's not supposed to do.

Testing is an extremely creative & intellectually challenging task. We must be aware of these classic testing mistakes to ensure we do our part in producing a quality product. To read the article in full, you can find a copy on Brian Marick's web site at the following URL: http://www.Testing.com/writings/classic/mistakes.html

The First Bug

It was the summer of 1945. The US Navy was rushing to finish Mark II, one of the first American large-scale digital computers.

"It was a hot summer with no air conditioning, so all the windows were open," wrote then Navy Captain Grace Hopper in the Annals of the History of Computing." Mark II stopped, and we were trying to get her going. We finally found the relay that had failed. Inside the relay -- and these were large relays -- was a moth that had been beaten to death by the relay. We got a pair of tweezers. Very carefully we took the moth out of the relay, put it in the logbook, and put scotch tape over it.

"Now, Commander Howard Aiken had a habit of coming into the room and saying 'Are you making any numbers?' From then on if we weren't making any numbers, we told him that we were debugging the computer. To the best of my knowledge that's where it started."

This logbook page, with the first computer bug still taped to it, is at the Naval Museum at the Naval Surface Weapons Center in Dahlgren, Virginia, USA.

-- The entire story, with a picture of the logbook and the moth taped into it, is recorded in the "Annals of the History of Computing," Vol. 3, No. 3 (July 1981), pp. 285--286.

From the Hacker's Dictionary..

Bug (n)

An unwanted and unintended property of a program or piece of hardware, esp. one that causes it to malfunction.

Antonym of feature.

Examples: "There's a bug in the editor: it writes things out backwards." "The system crashed because of a hardware bug." "Fred is a winner, but he has a few bugs" (i.e., Fred is a good guy, but he has a few personality problems).

Browse the rest of the Jargon File at: http://www.tuxedo.org/jargon

Dr. Seuss Quotes..

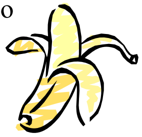
Oh, the things they will bump!
Oh, the things they will hit!
Oh, I do not like it!
Not one little bit!

UNLESS someone like you
cares a whole awful lot,
nothing is going to get better.
It's not.

Bug Fun Facts

Cockroaches & Lobsters are kissing cousins.

You're more likely to be a target for mosquitoes if you eat bananas.



Males of some giant water bugs carry the eggs on their backs until they hatch. Females hold the males and cement the eggs to them.

Ants stretch when they wake up. They also appear to yawn in a very human manner before taking up the tasks of the day.

An estimated 80% of animals on Earth have six legs, i.e., are insects. The more than 10 quintillion bugs fall into some 800,000 species.

Although "bug" has been used as a common name for any insect, this name was first used for species of Hemiptera, and these insects are considered the "true bugs."

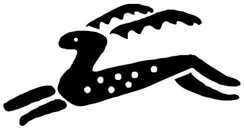
A dragonfly has a lifespan of twenty-four hours.

Worker bees are female.

And some Christmas Fun Facts..

Assuming Rudolph was in front, there are 40320 ways to arrange the other eight reindeer.

Reindeer like to eat bananas.



Reporting Bugs within the Software Environment

By Sherri Howell

Just as the purpose of testing is to find the most relevant bugs, the purpose of writing a bug report is to ensure that they are corrected in the software under test.

Good bug reporting is an essential part of achieving this. A good bug report is concise, persuasive, and informative. The more understandable a bug report is, the more likely that a programmer will want to deal with it. A good bug report should include clear directions on how to reproduce the defect. If no procedure is given, the bug is likely to be dismissed by developers.

To write a fully effective report you must ensure that you understand and can reproduce the problem yourself. The amount of time spent simplifying the procedure for reproducing a bug or exploring the ways that it could occur may save the programmer some time and the company some money. Otherwise, programmers may dismiss the report as irreproducible. If you can't reproduce a bug, admit that you can't reproduce it, and report it anyway. Say what you tried, and describe all error messages as fully as possible. It may be a symptom that may help to fully identify a related defect, so never toss out a report because you cannot reproduce the problem.

The seriousness of the bug must also be assessed. Don't make the common mistake of rating bugs less serious than they are (see review of Brian Marick's paper *"Classic Testing Mistakes"* elsewhere in this issue). Be persuasive about the priority of the bug. If it is serious, explain why a customer would view it as a problem.

As everyone knows, nobody likes to be criticised. Therefore, avoid describing problems in a way that makes the developer seem sloppy, or unprofessional. Developers and QA professionals share a common goal: they both want to present a product that is efficient and user friendly. If comments are insulting, they poison the relationship between developer and tester. [See Marick's other paper titled *"Working Effectively with Developers"* from his web site at: http://www.Testing.com/writings.html - *Ed.*]

TOP TEN HINTS FOR REPORTING BUGS*

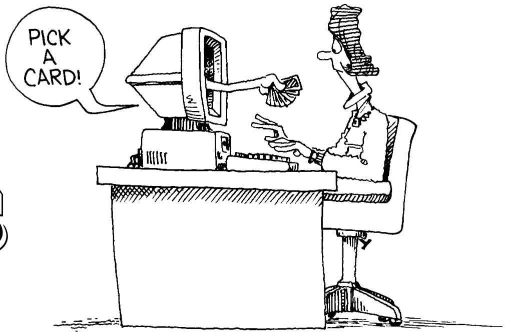
1. DON'T report a bug that you can only reproduce once.
2. ALWAYS examine the program's error handling procedures.
3. DO make sure that there are back up data files before attempting to replicate a bug that has corrupted data.
4. DO verify that your bug is not a side effect of another bug before reporting it as a new one.
5. DON'T be too quick to blame a bug on hardware. The problem is rarely the hardware.
6. DO check if the bug is time dependent, i.e. does special processing at a certain time of day.
7. DO keep a good relationship with the programmers
8. If you are testing on the fly and you find a problem that you can't repeat, you have probably forgotten about what you did.
9. NEVER leave your computer logged on if it is unattended
10. ALWAYS put a notice that testing is in progress on the computer you are testing when you leave it unattended.

*Source: "Testing Computer Software, 2nd ed.", by Kaner *et al*, 1993

Good bug reporting is the key to communicating within the software development cycle. Remember that how well you report a defect directly affects how likely the "powers that be" will fix it. Our ultimate goal is to ensure that the products and services that we test are reliable and perform their intended functions. Following these few guidelines will ensure that we do our jobs to the best of our ability.

TQM

TIPS & TRICKS



Susie **QA** is here for you, dear *TQM* reader, and wants to help answer any questions you have in your QA role. She will try to provide you with useful suggestions on how to make your job easier. Please forward any and all questions or useful tips to Susie QA at: news@kwsqa.org.

Q: What do I do when I find too many bugs too fast?

A: As a general rule: **Document all issues discovered while testing**. Good documentation will help when referring to a problem in the future. However, not all of the documentation has to be forwarded onto the developer or Team Leader. You may want to send only a summary and keep detailed information in a common, shared “testing results” folder. Do not throw out screen shots or step by step info, you will need to be able provide it upon request or to help you fill out problem reports for deferred issues later.

Most companies use a bug tracking system or tool to track all relevant issues discovered while testing. But if you are finding too many bugs too fast, entering issues into the bug tracking system may be time consuming. An Excel spreadsheet is an easy and effective way to quickly summarise issues. It may also be less overwhelming for the developer than sifting through a stack of papers to find the required information. All issues require a reference number, date, tester name, module identification, and a short but descriptive comment area. When the issues are fixed they can then be moved to a ‘fixed’ tab within the worksheet, and deferred issues can be entered into the tracking system. This way, everything is documented and can be easily recalled at a later date.

There is also the possibility that you are testing the wrong area of the software, and that the developers know that the functionality in that section is incomplete. Find out and make sure before the list of bugs gets too large so that you don't waste your time.

TQM

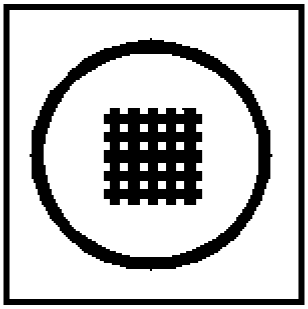
Resources on the W^3

Site	Strengths
STORM: Software Testing Online Resources: http://www.mtsu.edu/~storm/	A nexus of Software Testing Online Resources at MTSU, this site is designed to be a "first-stop" on the Web for software testing researchers and practitioners.
Automated Test Tools: http://www.sqa-test.com/toolpage.html	Automated Testing Specialists (ATS) publishes a fairly good list of available automated test tools, defect tracking & test management tools, test tool reviews and more.
BugNet: http://www.bugnet.com	BugNet is "the world's leading provider of PC software bug fixes." This site requires you to subscribe to gain access to the information and services.
Error Message Collection: http://www.errormessages.org	Ben Ezzell, author of *Developing Windows Error Messages*, has created an entire web site devoted to error messages.

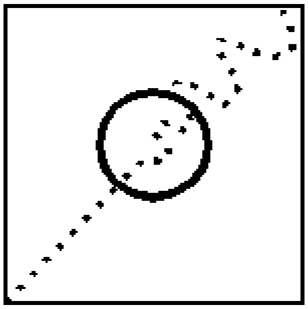
Droodles

Take your best shot at the following droodles. Suggested answers may be found elsewhere in this issue. Good Luck!

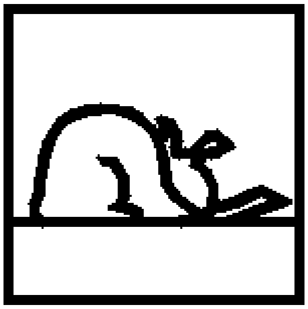
1.



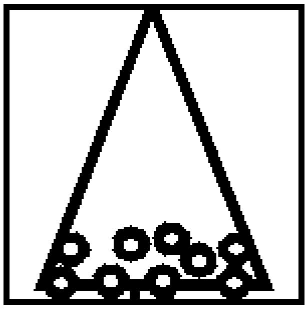
2.



3.



4.



Real Quotes

Remarks from IS Professionals to an internal group of Testing Consultants:

"I don't want you to teach Business Units any testing techniques, because then they would find more problems and create more work for us."

"We've done a good job of testing, we haven't found any bugs."

"We develop in C, we don't have to unit test."

"We won't have enough time to fix the bugs in production if we continue testing."

"We have to stop testing. If we find any more bugs, we won't meet our implementation date."

"My business unit said testing took too long and found too many bugs. They asked how much time it would save if no testing was done, and it was just put into production."

"I want the best [test] cases for the entire system to fit on one page."

"I don't have time to test, I'm too busy fixing bugs in production."

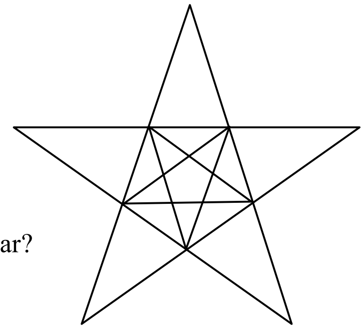
"I don't want to test because finding errors in testing reduces the morale of my group."

"The change that brought the system down was a two liner: it was low risk and wasn't worth testing."

QUIZZES & CHALLENGES

☆ = Very Easy, ☆☆ = Easy, ☆☆☆ = Average, ☆☆☆☆ = Difficult

Solutions to these problems will appear in the next issue. Enjoy!



1. What's Next? ☆

What is the next letter in this series?

O T T F F S S …

2. Star Gazing ☆☆

How many *triangles* can you find in this Christmas star?

3. Looks Okay ☆☆

Read the following paragraph and find out why it is unusual:

Obviously this writing is far from normal; it is sort of choppy and is not particularly fascinating, but don't mind that. It will all work out all right as soon as you know its solution. If at first you fail, just try again and again until it dawns on you. If you can't do it look at its solution.

4. Murder in a Small Software Company ☆☆☆☆ (only 3 ☆'s with hint)

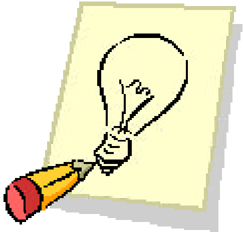
A system crash occurred one evening in a small company which consisted of four employees: one senior developer (male), one junior developer (female), one senior marketing person (female), and one junior tester (male). One employee intentionally crashed the system of another employee (the victim) resulting in a terrible loss of important data and information. The third employee witnessed the crime, and the fourth employee was an accessory after the fact.

1. The senior employees were older than the junior employees.
2. The accessory and the witness were of opposite sex.
3. The oldest employee and the witness were of opposite sex.
4. The youngest employee and the victim were of opposite sex.
5. The accessory was older than the victim.
6. The senior developer was the oldest member of the team.
7. The guilty party was not the youngest employee.

Which one of the four employees was the culprit?

Hint: Which role did the youngest employee play? Which employee was the youngest?

Solutions to Last Issue's Challenges



1. The Mule and the Donkey

The donkey had five and the mule seven sacks.

2. Two Cyclists

15 miles. (You didn't try to solve this one using brute force did you?)

3. Weight Lifters

(c) Viktor can outlift Boris by more than he can outlift Tam.

4. Wolves vs. Sheep

The solution at right shows the locations of the wolves (W) and sheep (S) in the grid.

Other solutions to this puzzle may exist which are mirror reflections of this one.

		W	W	
W			W	
W				
				S
	S			S

KWSQA Mailing List Information

The KWSQA offers a ***free*** mailing list to provide the means for software quality professionals to share information, ask questions, provide links to other Internet resources, and more. The mailing list is maintained using the eGroups.com web server, which provides such features as:

- Archive of past messages
- Files area to hold sample templates, presentation files, etc.
- Links to QA and Testing resources on the internet
- Online Polling feature for quick surveys
- Calendar of events
- and more!

To find out more about this networking resource, visit the site at:
http://www.egroups.com/group/kwsqa

To subscribe, send an email to:
kwsqa-subscribe@egroups.com

If you have any questions or comments, please send them to:
kwsqa-owner@egroups.com

Please note that this is a mailing list for sharing information only, and that job postings and solicitations are NOT allowed.

What is Quality?

Defining Quality reminds us of a story about three blind men who came across an elephant. The first one touches the tail and declares that it is a snake. The second one touches the elephant's leg, and claims that it is tree trunk. The third one touches the elephant's side and says that it is a wall.

When we attempt to define "Quality" we encounter the same type of situation. We approach the topic of quality with trepidation. Everyone knows what "Quality" is, but disagreements abound. Therefore, in attempting to answer this age-old question we invite our readers to contribute their opinions, theories, and considerations to our next issue of *The Quality Mark*.

Send in your replies via email to news@kwsqa.org and we will print them.

Droodle Answers

1.(A) Mended Doughnut
or
1.(B) Plate of Spaghetti
Served by Neat Waiter

2. Ants walking through
a drop of whiskey

3. The dog that eats
homework

4. Tall Christmas tree
decorated by short kids

About…

The Quality Mark

The Quality Mark is the official newsletter of the Kitchener-Waterloo Software Quality Association (KWSQA). It is published quarterly (in September, December, March, and June), and is distributed free to all local KWSQA members.

Challenge us to make a difference! *The Quality Mark* is about you and your interests. We are committed to making this newsletter a practical resource for you by responding to your suggestions and criticism. We invite you to give us feedback about what you find useful and what you don't. Send us your real-world ideas for future issues.

News@kwsqa.org

Editor: Paul Carvalho

Contributors:
Sherri Howell,
Terri Zuccherato,
Joanne Carvalho,
Paula Pfeifle

If you are interested in helping out with the next issue of *TQM* please drop us an email at the above address. No experience is required. We're looking for levity, imagination, and a desire to help create a useful, professional resource for the local high-tech community. Please let us know if you wish to contribute in any way.

Mark Your Calendar

For updates, visit: http://www.kwsqa.org/

The KWSQA Meetings usually occur on the last Wednesday of each month (except in December, July and August), and are held in the Communitech building on King Street in Waterloo. Finalised meeting details will be posted on our web site and sent to the KWSQA eGroups mailing list.

DECEMBER	Wonders of Winter Festival of Lights (Waterloo Park - until 2 Jan. 2001)	21 - *Winter begins* (Shortest day of the year)
	5 - STC General Meeting	25 - *Christmas Day*
	7-10 - Christkindlmarkt http://www.christkindlcanada.com/	31 - New Year's Eve Festival of the Night - Kitchener City Hall

JANUARY	1- *New Year's Day*	24, 25 - ASQ Quality Workshop (Bingeman Conference Facility)
	9 - STC General Meeting	31 - **KWSQA Meeting**
	22, 23 - ASQ's First Six Sigma Conference - San Diego, California http://www.asq.org	

FEBRUARY	6 - STC General Meeting	28 - **KWSQA Meeting**
	7 - ASQ Toyota Plant Tour	
	14 - *Valentine's Day*	

For more information on events happening in our local community, check out the following web sites:

- Local (KW) community events: http://www.kw-visitor.on.ca/
- Communitech seminars: http://www.communitech.org/
- Society for Technical Communication (STC): http://www.stc.waterloo.on.ca/
- ASQ Kitchener Section: http://www.asqkitchener.com/
- TASSQ events (in Toronto): http://www.tassq.org/

Know of any other great local links and/or events that we should add to our calendar? Let us know!