Words Words and more words…. OR Follow up on Lunch & Learn ~

“What’s a bug? How to tell the developer it really IS a bug, and when is the developer right?”

Let’s approach a subject carefully that I didn’t address in the L&L, how to tell a developer it really is a bug. The short answer to that is simple, “Politely with respect.” I have never forgotten, as a QA my basic job is to tell somebody they did something wrong, that I myself cannot do. **A little like telling a Marine Corps Scout Sniper, “You Missed.”**

Every single QA book, including “**How to break software**” written by **James Whittaker**. (below) They guy who first started the very first ever college level QA degreed program at Florida State, includes a small part about people skills. NEVER have I ever come across a Chapter, paragraph, or even a sentence in any of my programming books about people skills. So, if you get into a QA role, show respect. Trust me writing code is hard, be kind when you write up bugs.

NOW having said that, how do we tell a developer or anybody, “That really is a bug.” OR if asked a different way, “What are the five things that MUST be in all bug reports? HINT; REALLY BIG HINT HERE FOLKS. If I interview you for a QA job an you blow this, you can still get the gig. But if you read this and miss it, you missed a big clue.

Bug Reports, ALL, need five things.

* Title 🡺 Anything that when you scan it in a list of 1,000 titles tells you about the issue.
* Description 🡺 What happens when this fails, typically a paragraph.
* Replication Steps 🡺 How do you make it fail. Too much detail nobody reads, too little they can’t make it fail.
* The Expected Results 🡺 it should do X
* The Failure or Actual Results 🡺 It does do this.
* Six, if needed 🡺 A screen shot is worth 10,000 words in the software world.

**What isn’t a bug?** Our dear Kirsten Naughton brought to my attention something that is both NOT a bug, and Probably will never be fixed. Even if it does hit my personal yardstick of irritation to the CEO and student.

Let’s dig into this a little bit, bear with me I am going to use a lot of works and a few pictures.

Everybody thinks bugs are when a program, calculates wrong, performs wrong, costs money it should not, has security issues and flaws, or just doesn’t do what the end user wants.

I offered a couple of other ideas, (1) If it pisses off or embarrasses the CEO (2) If it pisses off or embarrassed the customer. I think these are bugs as well.

But, another question, how perfect is perfect? We all live with and accept a little bit of imperfection in our world. The very best HDMI cables from Best Buy are $1,200 for a 3 foot cable, my TV was $600. I like to say, ‘That dog won’t hunt.’ I will do nicely with the $50 cable thank you.

Change that a bit, I can buy a helmet for my motorcycle at Walmart for $75, or I can get a SNELL approved ARAI for $750. My brain goes in it! I’ll take the ARAI every day.

This issue I am about to describe falls into that category of, “We gotta cut this off and move on.” Even if it does hit my #1, 2 yard sticks for bugs. This code is not flawed. We could write this fix and raise the cost of the program by over $1,000 for each and every single question because of the greater coding time and testing time. Consider this!

Student A is taking Course 1011 (That’s B in binary) and they get to an exercise. Let’s pretend it’s Full Stack Web Online Course number 101 and maybe even Lesson #1.

Maybe it’s the Introduction to Front End Foundations

* Section ‘const vs. let’
  + Right after tip:
    - When using ES6, you should primarily use the const keyword to declare variables and let in special cases where reassignment may be necessary. If you are unsure, start with const, and if you need to reassign the variable directly, then you can change const to let.
* Did you all just catch that? I just gave you the replication steps that take the developer directly to the point in code with the issue.
  + And ~ Let’s say they get to the part where the student needs to enter in a bit of script. Like this
    - const numberFive = 5;
    - (That is exactly what we want the student to enter)
* Put your shoes in the developer’s feet at this point.
* YOU need to write code that is going to provide feedback to the student when they get it right, and when they get it wrong. You may even write something like this?
  + IF [Code Entered === ‘const numberFive = 5;’]
    - Send ‘Success’ to student;
  + ELSE
    - Send ‘Try Again’ to student
* That is called PSUDO CODE. It is human / English words that represent what we want the code to actually do.
* NOW ~ the concept we are trying to get across to the student is that the word ‘const’ followed by a word, and it can be literally just about any word. Heck it could have been fuzzyBunny for all the JavaScript processor cares.
  + const followed by a word
  + then the assignment operator ‘=’
  + then anything in this case the integer 5
  + then semicolon ;
* Will be the bit of code in JavaScript that creates in memory the variable with the name ‘numberFive’ and will assigned to that variable the value of 5.
* BIG POINT HERE ~ IN JavaScript is there any reason in the world these are not valid?
  + const fuzzyBunny = 7;
  + const numberFive = 7;
  + const EightyEight = 7;
  + const numberFiveIsAlive = 7;
  + const five = 7;
  + or
  + const numberFive = 5;

If my memory serves me correctly. Every single one of the above lines is valid script. However, that isn’t what the problem is.

Let’s put this problem a different way. YOU are a developer and YOU need to write something that is going to work 10,000,000 times perfectly every single time to teach the simple concept of assignment of a variable. So, I am going to bet you probably write something that looks like this.

if (numberFive === 5){

print("Success")

} else {

print("Try Again")

}

Now the student doesn’t see this. They see their window to the lesson that gives them a Boolean indication.

* ‘Success’
* ‘Try Again’

When you do this lesson, did you ever try making it say, “NumberFive”? I did I am a natural tester. But what does ‘NumberFive’ do verses numberFive, or even numberfive? Nothing, it never even gets to the ‘IF’ part of the LMS.

Typically, a student is going to look at that ‘Try Again’ and scroll up to the text above and see this.

* **const** thisNumber = 1234;

After reading it they will enter const numberFive =5; and receive the ‘Success’ message. GREAT! They go on.

Then comes this one, and this is the point where it becomes a big long wordy mess. Very little is really, ‘Wrong’ in code writing it’s better solutions and less good solutions.

The next project the student hits wants them to assign a string value to a variable by using the ‘let’ key word. This is a subtle variation on the time-honored tradition of ‘Hello World’.

Instructions read like this:

* Use the `let` keyword to assign the string `"Hello world"` to a variable named `greeting`.

Just like before put yourself in the shoes of our ( or any ) development staff. All you want to teach them is to write a single simple line of code that looks like this …. let greeting = “Hello world”;

* Let Keyword :: variable Name :: assignment Operator :: String Variable :: Final Closing ending semi-colon.

If this was a face to face interaction with an instructor who is holding your hand, can you enter this?

* let x = “Dude”;
* YES
* Is the concept any different?
* NO

Test for the 250,000,000 or more possible variations on the concept of

* Let :: Variable :: Assignment :: String :: Semi-colon ?
* OR
* Test for the most common exact match, and move on?

Is it possible to write code that would first look for the ‘let’ key word? YES

Then look for a variable that is legal? Not just the requested one of ‘greeting’ but any variable that matches the JavaScript language requirements? YES. What the heck does Scott mean by that? I would be that this isn’t a legal Javascript line of code

* let let = “Hello world”; (Why isn’t that legal?) Or at least smart? Because ‘let’ is a keyword.

Yeah but we are not talking about just the keyword here Scott, the information between the quotation marks are string literals and can truly be anything. YES

SO in concept these too are identical.

1. let myNewVariable = “Hello world”;
2. let myNewVariable = “Hello World”;
3. let myNewVariable = “HelloWorld”;
4. let myNewVariable = “Hell World”;
5. let myNewVariable = “Hello Worlds”;

Let me show you in psudo code that that would look like. It is going to look something like this for only the 5 listed.

IF(myNewVariable === "Hello world"){

print("Success")

}

ELSE IF (myNewVariable === "Hello World"){

print("Success")

}

ELSE IF (myNewVariable === "HelloWorld"){

print("Success")

}

ELSE IF (myNewVariable === "Hell World"){

print("Success")

}

ELSE IF (myNewVariable === "Hello Worlds"){

print("Success")

} ELSE {

print("Try Again")

}

* let’s get back into our developer shoes. Think about how many possible variations there are. It truly is infinite in the number of possibilities.
* That big nasty string only took care of testing for a few minor variations, ‘W’ and ‘s’ and ‘missing space’ and ‘no-o’ in the student’s code.
* There is an argument to be made, “yes bug ‘Hello World’ and ‘Hello world’ are essentially the same.
  + To a human, but tell me again how to make a PB&J sandwich? We have all watched that video.

We want the student to write this ~ let myNewVar = “Hello world”;

We only test the variable name IF and ONLY IF it is ‘myNewVar’ anything else and we don’t even bother to test it.

We look to see if it matches what we asked, ‘Hello world’ we can only provide 1 answer per test.

* Success
* Try again

How many times or lines of code to you want to write to test what is asking the student to write 27 characters?

In reality they could write as few as 10 ~ let x=”y”; Even if that was all we asked it would be 26 factorial variations.

* BTW 26 factorial is this big.
  + 2 Factorial = 2
  + 4 Factorial = 24
  + 7 Factorial = 5040
  + 10 Factorial is = 3.6288 \* 10E6 HUM… 3,628,800
  + 4.032914611 \* E26 or 403291461126605635584000000
* To test accurately a student being asked to write just a single letter variable name combined with a single letter variable is a boat load of options.

Is this a bug?

* HUM ~ does it frustrate the end user, YES.
* Humm ~ OK has it frustrated our CEO, YES.
* AS a developer do I think the student should have read or watched the part about following directions better? YES.
* So again I’m going to ask, “Is it REALLY a bug? Or a student learning opportunity?”
* Still think it’s a bug, OK start writing a test for every variation of let x= “a”; just lower case letters

Really this is a lesson in where do we, cut off, or end things?

I would say right here. By now you have read 5 pages, and almost 2,000 words, and over 10,000 characters because of the difference between the ‘W’ and the ‘w’ in the word world in a single lesson. Because the student didn’t type in world they typed in ‘World’.

As I end let me be perfectly clear, I am in 100% sympathy with the student. Life can be frustrating, but at some point, we need to launch the space shuttle.

BTW according to Bruce S. in his book ‘Secrets and lies’ the space shuttle launches with 7 known bugs. At some point we need to stop writing code and launch the thing.

Secrets and Lies by Bruce Schneier

* <https://www.amazon.com/Secrets-Lies-Digital-Security-Networked/dp/1119092434>

How to Break Software by James Whittaker

* <https://www.amazon.com/How-Break-Software-Practical-Testing/dp/0201796198/ref=asap_bc?ie=UTF8>