LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-JRNL-211966

"Test" is a Four Letter Word

G. M. Pope

May 3, 2005

Better Software Magazine

Updated October 14, 2003

"Test" is a Four Letter Word

By Gregory M. Pope

12/28/03 v 1.0

For a number of years I had the pleasure of teaching Testing Seminars all over the world and meeting and learning from others in our field. Over a twelve year period, I always asked the following questions to Software Developers, Test Engineers, and Managers who took my two or three day seminar on Software Testing:

"When was the first time you heard the word test?"
"Where were you when you first heard the word test"?
"Who said the word test"?
"How did the word test make you feel"?

Most of the thousands of responses were similar to "It was my third grade teacher at school, and I felt nervous and afraid."

Now there were a few exceptions like "It was my third grade teacher, and I was happy and excited to show how smart I was."

But by and large, my informal survey found that "testing" is a word to which most people attach negative meanings, based on its historical context. So why is this important to those of us in the software development business? Because I have found that a preponderance of software developers do not get real excited about hearing that the software they just wrote is going to be "tested" by the Test Group. Typical reactions I have heard over the years run from:

"I'm sure there is nothing wrong with the software, so go ahead and test it, better you find defects than our customers."

to these extremes:

"There is no need to test my software because there is nothing wrong with it."

"You are not qualified to test my software because you don't know as much as I do about it."

"If any Test Engineers come into our office again to test our software we will throw them through the third floor window."

So why is there such a strong negative reaction to testing? It is primitive. It goes back to grade school for many of us. It is a negative word that congers up negative emotions. In other words, "test" is a four letter word. How many of us associate "Joy" with "Test?" Not many. It is hard for most of us to reprogram associations learned at an early age.

So what can we do about it (short of hypnotic therapy for software developers)?

Well one concept I have used (and still use) is to not call testing "testing." Call it something else. Ever wonder why most of the Independent Software Testing groups are called Software Quality Assurance groups? Now you know. Software Quality Assurance is not such a negatively charged phrase, even though Software Quality Assurance is much more than simply testing.

It was a real blessing when the concept of Validation and Verification came about for software. Now I define Validation to mean assuring that the product produced does the right thing (usually what the customer wants it to do), and verification means that the product was built the right way (in accordance with some good design principles and practices). So I have deliberately called the System Test Group the Verification and Validation Group, or V&V Group, as a way of avoiding the negative image problem. I remember once having a conversation with a developer colleague who said, in the heat of battle, that it was fine to V&V his code, just don't test it! Once again V&V includes many things besides testing, but it just doesn't sound like an onerous thing to do to software.

In my current job, working at a highly regarded national laboratory with world renowned physicists, I have again encountered the negativity about testing software. Except here they don't take kindly to Software Quality Assurance or Software Verification and Validation either. After all, software is just a trivial tool to automate algorithms that implement physics models. Testing, SQA, and V&V take time and get in the way of completing ground breaking science experiments. So I have again had to change the name of software testing to something less negative in the physics world. I found (the hard way) that if I requested more time to do software experimentation, the physicist's resistance melted. And so the conversation continues, "We have time to run more software experiments. Just don't waste any time testing the software!"

In case the concept of not calling testing "testing" appeals to you, and there may be an opportunity for you to take the sting out of the name at your place of employment, I have compiled a table of things that testing could be called besides "testing." Of course we can embellish this by adding some good sounding prefixes and suffixes also. To come up with alternate names for testing, pick a word from columns A, B, and C in the table below. For instance Unified Acceptance Trials (A2,B7,C3) or Tailored Observational Demonstration (A6,B5,C5) or Agile Criteria Scoring (A3,B8,C8) or Rapid Requirement Proof (A1,B9,C7) or Satisfaction Assurance (B10,C1). You can probably think of some additional combinations appropriate for your industry. Just don't call it testing!

Don't Call It Testing Table

A	B	C
1. Rapid	1. Quality	1. Assurance
2. Unified	2. Verification (and)	2. Validation
3. Agile	3. Experimental	3. Trails
4. Meta	4. Examination	4. Study
5. Flexible	5. Observational	5. Demonstration
6. Tailored	6. Conceptual	6. Prediction
7. Scalable	7. Acceptance	7. Proof
8. Integrated	8. Criterion	8. Scoring
9. Independent	9. Requirement	
10. Observed	10. Satisfaction	
11. Customer Based		
12. <none>		

So now that there are possible alternatives to using the negative word "test," we still have to deal with what to call the situation that exists when the expected result of running the test (experiment, study, examination, etc.) does not agree with the actual result. Common words currently used to describe this situation are "bug," "defect," "failure," "fault," or the somewhat less judgmental "incident." However, it may be wise to look at some additional alternatives. These names can also be useful for annotating "defect" reports and findings using automated tracking tools. For instance we could report a Potential Anomaly (A1, B1) or Unstable Believability (A6, B3) or Irregular Correctitude (A7, B2) or Suspect Convergence (A2, B5) or Fuzzy Correlation (A10, B6) or a Biased Presentation (A11, B11). For medical equipment a *Correctus Minimus* may be appropriate, and for software used in biological applications the *Hemiptera Heteroptera* (Latin for "bug") might have a nice ring.

Don't Call It a Bug Table

A	B
1. Potential	1. Anomaly
2. Suspect	2. Correctness
3. Tentative	3. Believability
4. Pseudo	4. Certainty
5. Unresolved	5. Convergence
6. Unstable	6. Correlation
7. Irregular	7. Correctitude
8. Arbitrary	8. Correspondence
9. Random	9. Censure
10. Fuzzy	10. Result
11. Biased	11. Presentation

Perhaps you can add to the “Don’t Call It A Bug” table, as well.

With a little creativity, those of us who specialize in this craft with the negative connotation can have some fun coming up with more palatable terminology. In the not too distant future we may hear conversations as follows:

“The software was so good that the developers felt it to be without bugs and not necessary to test. We did, however, perform some Rapid Requirement Proofs and found a number of cases of Irregular Convergence and Biased Believability. These findings were handled by the developers as trivial enhancements, which have now been fully implemented, and we are ready to ship after performing the mandatory Independent Observational Scoring.”

Good Luck.