MTBN.NET PLR Library
Category: Computers_Technology File: False_Failures__Worse_Than_Real_Failures_utf8.txt
Text and Word PLR Article Packs available at PLRImporter.Com

Title:
False Failures: Worse Than Real Failures

Word Count:
452

Summary:
Better to fail for real than fail to really fail. Huh?
We know you've experienced this. Let's say you just added some new functionality into your software, and you run a new build. And let's say that 50% of your test cases fail. What is the first thing you assume?

We've asked this same question as our "teaser pitch" last winter to 100 developers and QA professionals who walked up to our booth at a recent conference, and 95 of them had the same answer! The tests must be br...

Keywords:
soa test, service oriented architecture, testing,software,quality assurance,qa,j2ee,soap,java,.net

Article Body:
Better to fail for real than fail to really fail. Huh?
We know you've experienced this. Let's say you just added some new functionality into your software, and you run a new build. And let's say that 50% of your test cases fail. What is the first thing you assume?

We've asked this same question as our "teaser pitch" last winter to 100 developers and QA professionals who walked up to our booth at a recent conference, and 95 of them had the same answer! The tests must be broken!

This creates a cascading set of bad assumptions that will make your manager recite the adage about "ASS out of U and ME" on the whiteboard at the next project meeting. Here's why.

* You assume that the problem is not with your application, it is with the test cases themselves being broken or no longer valid.
* So you spend time comparing the test cases with whatever changed in your new build.
* Then you dig into the test scripts to try to figure out why the test case is no longer passing, and rework them until they pass.
* Or you just give up and try validating by clicking through your old Word

MTBN.NET PLR Library
Category: Computers_Technology File: False_Failures__Worse_Than_Real_Failures_utf8.txt
Text and Word PLR Article Packs available at PLRImporter.Com

document test cases. Fun busy work.

How can you possibly call this testing? Rather than using the test to validate the application, you are using the application to test the test case - which is a program you coded!

Yes, unit tests are important for finding structural bugs in your code. But once a unit test tries to get beyond that granular level of testing, it becomes another fragile program in your development environment.

It is outrageous to assume that relying on coded unit test cases alone offers you any value in functional testing. In fact, the whole process is so manual and highly inefficient, that you wonder if you are doing anything more than making busy work for your own team.

Unit testing has its limits. There are methods people have tried to get beyond these limits, but it is like challenging the theory of gravity.

* Attempting to code for reuse - may seem possible but can only get you to the edge of Unit testing's limits.
* Attempting to test the UI with your QA group, doesn't really work if you can't see those middle and back-end layers.

What makes false failures so dangerous? Besides the fact that they are a morale vampire that will make the team give up on testing, false failures impact the overall effectiveness of testing. If you don't know if a failing test case is even valid, what do you really learn from testing? It is like a detective that never gathers evidence.

Time to declare war on false failures.