

Re: TDD: Test-Driven Design or Test-Driven Development?

Source: <http://coding.derkeiler.com/Archive/General/comp.object/2004-02/0388.html>

From: Phlip (phlip_cpp_at_yahoo.com)

Date: 02/03/04

Date: Tue, 03 Feb 2004 13:06:58 GMT

Vladimir wrote:

> *I will never get tired of repeating...*
>
> *If you intent to get tests pass you'll succeed. If you then approach*
> *software with intention to make it fail you'll succeed again. Wow!*
Amazing!
> :)

"If you are curious, or code does something unexpected, or you receive a bug report, always write a new test. Then use what you learned to improve design, and write more tests of this category. If you treat the situation 'this code does not yet have that ability' as a kind of bug, then the TDD cycle is nothing but a specialization of the principle "capture bugs with tests".

"But this power creates new risks. If all your tests request bugs, if you leave huge matching gaps in testage and code, and if you never refactor your code but leave it crufty, frequent testing will also aggressively support these activities, and provide a very high apparent velocity.

"Teach your colleagues to write tests on your code, and learn to write tests on theirs too. If they write a test that fails due to missing abilities, not faults in deployed abilities, treat the failing test as a feature request, and get it prioritized."

You are playing the familiar "declaim one practice in isolation from the others" game. The White Book also covers that.

--

Phlip

<http://www.xpsd.org/cgi-bin/wiki/TestFirstUserInterfaces>