

Re: how many bugs do you find and correct during TDD?

Source: <http://coding.derkeiler.com/Archive/General/comp.object/2004-12/0553.html>

From: Tammy (qqq_at_ppppp.com)

Date: 12/31/04

Date: Thu, 30 Dec 2004 15:37:35 -0800

Steve Jorgensen wrote:

> So, let's say the average test/fail/pass cycle is 10 minutes. In that time,
> first you write a test you expect to fail. Perhaps, it does fail, and perhaps
> it doesn't. If it is not as expected, is that a bug, an improperly written
> test, or is that a misunderstanding of what the code's current untested
> capabilities are. A little bit of digging is done, and it is found that the
> test was not written properly. Is that a bug?

Yes.

> Next, the test is corrected,
> and now fails as expected. That's not a bug because we expected it, right?

Agreed.

> OK, now we do the simplest thing that could possibly work to make the test
> pass, and the test still fails (or another test fails). Is that a bug? I
> dunno – we didn't know whether to expect the test to pass or fail because we
> were experimenting to see if the simplest thing actually did work.

If you expected it to pass and it didn't then that is a bug.

> As you can see from this pattern, TDD is not so much about finding bugs before
> they go into production as setting up tests that ensure code that passes them
> functions as designed, then making the code meet the tests. I would say that
> a bug, in this context is defined as something that got through our net either
> because we forgot a test, wrote an inadequate test, we misunderstood the
> request, or the request itself was the result of a misunderstanding of the
> problem. How many bugs does TDD catch then?

I am trying to figure out the error rate of a human programmer as they work. So only classifying a bug as something that slips the net doesn't accomplish that goal.