Re: Test Driven Development

Source: http://coding.derkeiler.com/Archive/General/comp.object/2003−12/0573.html

---

From: Harry Erwin (*herwin_at_theworld.com*)
Date: 12/09/03

Date: Tue, 9 Dec 2003 10:28:12 +0000

H. S. Lahman <h.lahman@verizon.net> wrote:

> Responding to Erwin...
>
> We seem to have different definitions about what 'testable' means. In
> this context I believe it means that one can make a measurement to
> unambiguously determine whether the implementation has correctly
> satisfied the requirement.
>
> When defining those scenarios, how does one know that the only useful
> test scenario (i.e., where pass/fail is meaningful insofar as whether
> the requirement is satisfied) is 750/250? That is, the test result for
> any other mix is ambiguous. There is nothing in the statement of the
> requirement to indicate that.
>
> IOW, the requirement identifies a statistical measurement without
> defining how it relates to correctness of the implementation. My point
> is that for a statistical requirement like the example to be testable,
> it must also define the sampling criteria so that a pass/fail result for
> a given test case unambiguously determines whether the requirement is
> satisfied (pass) or not (fail).
>

That's an old argument that I can remember having in the mid−70s. I suppose you can substitute 'demonstrable' for 'testable'. We would measure the software performance at the task level and in the real−time operating system and then do a very sophisticated queueing analysis (See Chapter 3 of Volume 2 of Kleinrock for the theory) to demonstrate that the system could handle all realistic loads. (The invention of the spreadsheet was a godsend for this−−up until then, we had used specialized programs to do the same thing.) This model was experimentally calibrated to system performance in integration tests to demonstrate its validity.

On Site Defense, we also came up with a series of five overload scenarios that the customer agreed would be adequate to demonstrate that the system met it's performance requirements, since they all exceeded anything realistic. Of course, during operations, one of those proceeded

to happen. We handled it OK.

The gist of my argument is that a deep knowledge of statistics and queueing theory can give you some sharp tools as long as the system is fairly well−behaved. And then in 1989, I proceeded to demonstrate that you could get chaotic performance in computer systems if your dynamics were non−linear and your load high enough. (Erwin, 1989, "Mixing and Sensitive Dependence on Initial Conditions in Computer Systems," Computer Measurement Group Transactions, 65:3−6, Summer 1989.) That paper is actually an argument for your position.

−−
Harry Erwin <http://www.theworld.com/~herwin/>