

Re: Art of Unit Testing

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-08/msg00875.html>

- *From:* Peter Hansen <peter@xxxxxxxxxxx>
 - *Date:* Wed, 03 Aug 2005 09:51:49 -0400
-

phil hunt wrote:

On Tue, 02 Aug 2005 21:26:28 +0200, Christoph Zwerschke <cito@xxxxxxxxxxx> wrote:

According to the "extreme programming" paradigm, testing should be done several times a day. So a requirement for extreme programming is that tests are fast enough. If the testing needs too much time, people are discouraged to test often.

Indeed. Running the tests should ideally take less than a few seconds. Any longer, and people won't use them so often.

That's probably too general a statement to be very useful, and I don't think it's probably helpful to a newcomer who is seeking guidance.

If one makes the XP-style distinction between "unit" and "acceptance" tests, then the acceptance test **suite** should take no more than some number of minutes. I believe ten minutes absolute max is a commonly offered rule of thumb, but I've found even eight minutes a little too long. Five seems to be just fine, and less if of course wonderful.

For unit tests, the entire suite should take only a minute or two, max, if possible, and preferably well under a minute. If you're doing test-driven development (TDD), you will want to be able to run unit tests (though maybe not all of them) sometimes a couple of times per minute! If the overhead is fifteen seconds, you can't type much new code in between running the tests.

Still, in a large project (and especially one written in Python, with the overhead of interpreter startup and the cost of executing bytecode) the suite can get fairly long if you have many hundreds of tests. In that case, there's nothing that says you can't run individual test files, or even individual test cases, when you are focused on one small area of the code. That should let you run tests in only a couple of seconds, as phil just recommended, in almost any case.

Re: Art of Unit Testing

Having your code organized nicely into packages can help as well. If you have a helpful "test runner" utility which scans subdirectories for test cases, you can easily run all the tests in a given package independently of the rest of the app while you are working on that one package, reducing the risk inherent in running only a subset of your full test suite.

-Peter

.