

Some Benefits of Unit Testing

Posted At : September 10, 2008 6:32 AM | Posted By : Bob Silverberg

Related Categories: OO Design, MXUnit, ColdFusion, TDD

I must start by saying that, as much as I like option groups, I *love* Unit Testing. I had written previously that [I Like Unit Testing](#), but that brief infatuation has blossomed into a full blown love affair. I have two people/groups to thank for that, [Paul Marcotte](#) and [the MXUnit Team](#), who produce an excellent suite of unit testing tools for ColdFusion.

I met with Paul in the Toronto airport a couple of weeks ago during a brief stopover (his, not mine) and he walked me through his current methodology for unit testing. He really cleared up some areas of confusion for me, and essentially handed over a set of practices that makes everything oh so easy. He's been blogging about his technique [in a series](#) that I heartily recommend.

And what can I say about [MXUnit](#) that hasn't already been said? It totally rocks! The Eclipse Plugin is da bomb! If you haven't checked it out, do so. Now. 'Nuff said.

So when do I get to hear about "Some Benefits of Unit Testing", you may be asking? Read on.

This post was actually inspired by [a conversation](#) on the [MXUnit Google Group](#). They were discussing the difficulty of making inroads with TDD in a corporate culture that undervalues "quality" software. This made me think about that fact that, as a one-man shop, I don't really face that difficulty, but also about the fact that I see other benefits of unit testing, over and above that of producing quality software. Some of these benefits may be due to the fact that I've only been attempting to change my development style to a more OO approach over the past year or so, so they may not be relevant to everyone, but here goes:

- I find that maintenance, whether it be bug fixes or incremental enhancements, can be completed much faster when I have a pre-existing set of unit tests for a component. Of course one has to take the time to produce those tests at the beginning, but even if "quality" isn't seen as of extreme importance, most shops realize that a tremendous amount of time, and therefore money, is spent on maintenance activities.
- I find that it helps greatly in both refactoring existing code and in writing new OO code from scratch. This may just be me being a newb to OO development, but I find that writing out the unit tests often helps me figure out how to structure my model. In the process of writing a unit test, I may realize that a method is doing too much and should be subdivided, or even that a method belongs in a different component. If a method seems very difficult to test I often find it is because my initial design isn't perfect, so writing the test gives me clues about how to restructure things.
- I find that I am actually able to develop faster, using the [MXUnit](#) plugin, than I did before. The fact that I can change as much code as I want and then click a button and be told whether or not my code changes worked as expected is fabulous. I believe that the time that it takes to run a page in a browser, and then visually interpret the results, over and over again, is greater than the time that it takes to write the tests themselves.

And then, of course, there's the fact that you will ultimately produce higher quality software ;-)