Client Visible Unit Tests

Posted At : July 28, 2009 9:00 AM | Posted By : Steve
Related Categories: Testing

We recently started working on a project that, while relatively small now, has the stated potential to grow into a rather large and complex system in the near future. On a few big projects that I have now, I have regretted not having used **TDD** on them from the beginning (having learned TDD only well after those projects were already under way).

This seemed a perfect opportunity to do now what I wished I had done earlier on those projects.

When I first started using unit tests, I thought I needed to write a unit test for every method in my system. That pretty quickly had the feeling of tedium. Most of my code runs off of **DataMgr** (or a further abstraction thereof) and is created a by a code generator. At first, I had the generator create unit tests for the generated code. Something seemed amiss.

It occurred to me only recently (perhaps I am a bit slow) that what I need to test isn't *methods*, but *business rules*. This was a big shift for me. This thought brought me to another issue I have had from time to time.

Sometimes a client will call me and ask what will happen if they take a certain action. This is a bad sign as it means that I have failed to make the behavior of the system as clear as it should be. I wanted a good way to document all of those (non-obvious) behaviors for my clients.

They didn't need documentation that told them that when they hit submit while editing a record that their changes would be saved. That stuff is obvious. They need the business rules documented.

Suddenly, these two problems appeared to be one.

So, I wrote out my business rules for the site (just four of them for now) and wrote my unit test stubs. Here is an example:

```
<cffunction name="shouldnotEnrollFullClass" access="public" returntype="void" output="no" hint="Don't allow students to enroll in a class that is already full (has as many enrollments as ""Max Class Size"").">

   <cfset fail("This test isn't implemented yet.")>

</cffunction>
```

These tests are written against **MXUnit**, but would work just as well using **CFUnit** or **cfcUnit**. Note that I followed the should/shouldnot prefix pattern of **BDD**. I also put the business rule itself as the hint for the test method.

Then I wrote a small program to find business rule unit tests and show them to the user.

After that, I finished writing the unit tests and then wrote the code to make them pass.

Now, my clients can see the business rules driving the system. They can even test them out themselves if they have any doubt that they are still working.

This is a new idea for me, but I think it has promise. If you have any experience with a similar approach, I would love a heads up on potential advantages or pitfalls.

- **Test Driven Development with ColdFusion Part I : An Introduction to Unit Testing**
- **Articles on TDD and ColdFusion on Adobe Developer Connection**