



## Code the Unit Test First

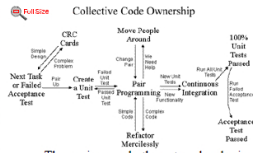
*Lessons Learned*

When you create your tests first, before the code, you will find it much easier and faster to create your code. The combined time it takes to create a unit test and create some code to make it pass is about the same as just coding it up straight away. But, if you already have the unit tests you don't need to create them after the code saving you some time now and lots later.

Creating a unit test helps a developer to really consider what needs to be done. Requirements are nailed down firmly by tests. There can be no misunderstanding a specification written in the form of executable code.

You also have immediate feedback while you work. It is often not clear when a developer has finished all the necessary functionality. Scope creep can occur as extensions and error conditions are considered. If we create our unit tests first then we know when we are done; the unit tests all run.

There is also a benefit to system design. It is often very difficult to unit test some software systems. These systems are typically built code first and testing second, often by a different team entirely. By creating tests first your design will be influenced by a desire to test everything of value to your customer. Your design will reflect this by being easier to test.



There is a rhythm to developing software unit test first. You create one test to define some small aspect of the problem at hand. Then you create the **simplest code** that will make that test pass. Then you create a second test. Now you add to the code you just created to make this new test pass, but no more! Not until you have yet a third test. You continue until there is nothing left to test. The coffee maker problem shows an **example written in Java**.

The code you will create is simple and concise, implementing only the features you wanted. Other developers can see how to use this new code by browsing the tests. Input whose results are undefined will be conspicuously absent from the test suite.



[ExtremeProgramming.org home](#) | [XP Rules](#) | [Pair Programming](#) | [About the Author](#)

Copyright 2000 Don Wells. All rights reserved.

