

What is Test Driven Development?

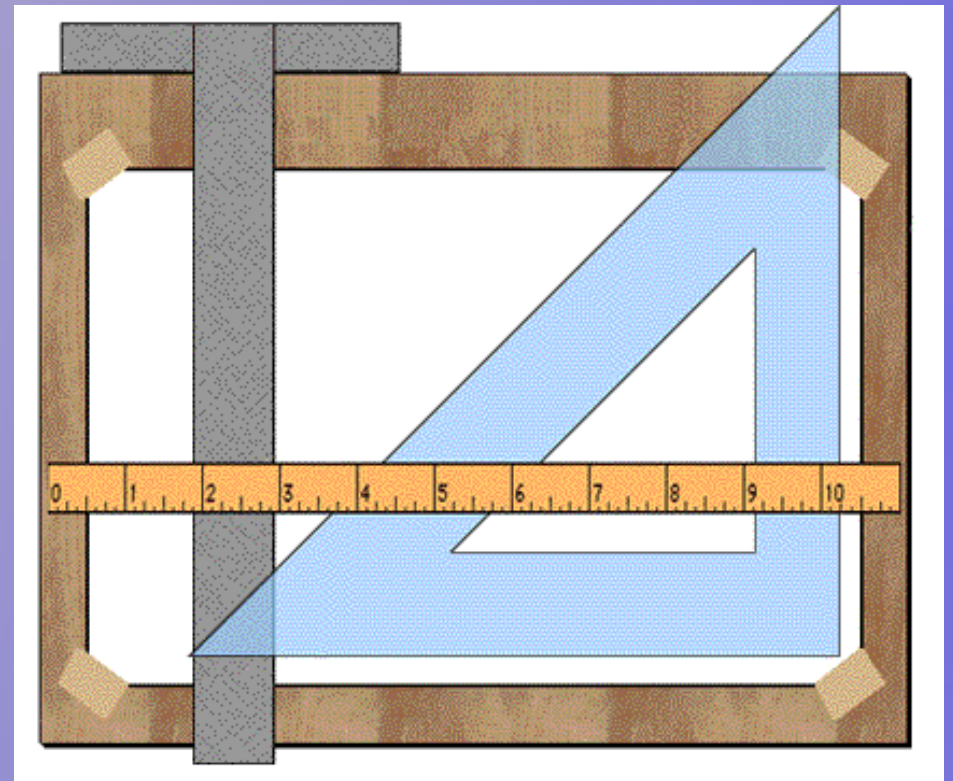
*"That's what a ship is, you know
It's not just a keel and hull
and a deck and sails.
That's what a ship needs.
But what a ship is... is freedom."*

-- Captain Jack Sparrow



TDD is a design tool

- The resulting regression suite is a side-effect, for free.
- Tests are a second client of the code, promoting reusability and flexible code.



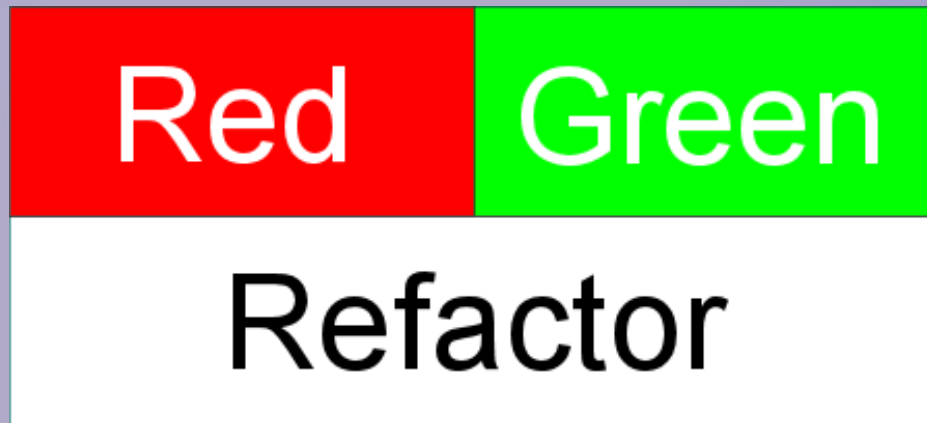
Three little steps

- Write a small test
 - Express what you want the code to do next
- Make it so
 - Do the simplest thing to make that test pass
- Make it clean and simple
 - Good code is flexible and extensible

Repeat until done



The Cycle



- Red
 - See the test fail
- Green
 - See the test pass
- Refactor
 - Improve the code
 - (and the tests)



Kent Beck's Simple Design

1. Run all the tests
2. Contains no duplicate code
3. Expresses all the ideas the author wants to express
4. Minimize classes and methods



Types of Tests

- Programmer Tests vs Customer Tests
- Fast Tests vs Slow Tests



A test is not a unit test if:

- It talks to the database
- It communicates across the network
- It touches the file system
- It can't run at the same time as any of your other unit tests
- You have to do special things to your environment (such as editing config files) to run it.

Michael Feathers

<http://www.artima.com/weblogs/viewpost.jsp?thread=126923>



As Brian Marick would say...



Books on Test Driven Development

- Test Driven Development: By Example
 - Kent Beck
- Test Driven Development: A Practical Guide
 - Dave Astels
- JUnit Recipes: Practical Methods for Programmer Testing
 - J.B. Rainsberger

... and numerous others



Books on Refactoring

- Refactoring: Improving the Design of Existing Code
 - Martin Fowler
- Refactoring to Patterns
 - Joshua Kerievsky
- Refactoring Workbook
 - William Wake

... and others



Bio

- a Software Development Consultant and Coach
- over twenty years of experience ranging from small embedded systems to corporate enterprise systems
- strong interest in lifelong learning
- more effective ways of creating software at the technical, interpersonal and organizational levels.
- I help teams learn more effective software development techniques while accomplishing their current projects.

