Unit Testing is for Weenies.
Unit Testing is for Weenies.

Important lessons learned from being a bullheaded developer...
Why should I spend time testing my code?
Why should I spend time testing my code?

* How do you know that it works, particularly if it’s part of a big system?
Why should I spend time testing my code?

* How do you know that it works, particularly if it’s part of a big system?
* How do you know which part is broken when it breaks?
Why should I spend time testing my code?

* How do you know that it works, particularly if it’s part of a big system?

* How do you know *which* part is broken when it breaks?

* How do you *know* that you know which part is broken?
I always test my code...
I always test my code...

```
$newThing = new Thing();
$newThing->doSomething();
$newThing->doSomething($else);
if ( $newThing->something == $else ) {
    echo "All done!"
}
```
I always test my code...

* How can anyone ever repeat that test?
I always test my code...

* How can anyone ever repeat that test?
* What did it test, anyway?
I always test my code...

- How can anyone ever repeat that test?
- What did it test, anyway?
- What do you have to show for the effort?
Testing is so repetitive...
Testing is so repetitive...
Testing is so repetitive...
Testing is so repetitive...

* Unit Tests test Objects.
Testing is so repetitive...

- Unit Tests test Objects.
- Objects have Inheritance.
Testing is so repetitive...

* Unit Tests test Objects.
* Objects have Inheritance.
* Unit Tests are themselves Objects.
Testing is so repetitive...

* Unit Tests test Objects.
* Objects have Inheritance.
* Unit Tests are themselves Objects.
* Don’t Repeat Yourself.
There just isn't enough time for testing!
There just isn't enough time for testing!

* How much time do we spend...
There just isn't enough time for testing!

* How much time do we spend...
* Debugging a script or application?
There just isn’t enough time for testing!

* How much time do we spend...
  * Debugging a script or application?
  * Finding the actual bug?
There just isn’t enough time for testing!

- How much time do we spend...
  - Debugging a script or application?
  - Finding the actual bug?
- How many billable hours do we lose?
There just isn’t enough time for testing!

* How much time do we spend...
* Debugging a script or application?
* Finding the actual bug?
* How many billable hours do we lose?
* Don’t we make our own schedules?
How to Sell Your Boss on Unit Testing
How to Sell Your Boss on Unit Testing

It’s not me, it’s the psycho I work for...
The Essence of Quality Control is Testing
The Essence of Quality Control is Testing

* Testing adds value to work and products
The Essence of Quality Control is Testing

- Testing adds value to work and products
- Lessons from other industries...
The Essence of Quality Control is Testing

* Testing adds value to work and products
* Lessons from other industries...
* Toyota makes Good Automobiles.
The Essence of Quality Control is Testing

* Testing adds value to work and products
* Lessons from other industries...
  * Toyota makes Good Automobiles.
  * Sam Adams makes Good Beer.
The Essence of Quality Control is Testing

* Testing adds value to work and products
* Lessons from other industries...
  * Toyota makes Good Automobiles.
  * Sam Adams makes Good Beer.
* Unit Testing is Quality Control.
Quality Control Testing MUST BE
Quality Control Testing **MUST BE**

* **Rigorous**, by testing everything,
* **Repeatable**, via automated tests,
* **Producible** upon demand,

Rigorous, repeatable, producible tests generate a more stable code base and a more accountable project.
Testing actually saves time and money

* Spend less time on tail-end debugging and finding those bugs.
Testing actually saves time and money

* Spend less time on tail-end debugging and finding those bugs,

* Expose logical flaws and assumptions that we all unintentionally build on,
Testing actually saves time and money

* Spend less time on tail-end debugging and finding those bugs,
* Expose logical flaws and assumptions that we all unintentionally build on,
* Build dependencies that work since they’re based on proven assumptions.
Topics to ask Google and Wikipedia about...
Topics to ask Google and Wikipedia about...

* JUnit - The granddaddy of 'em all.
* phpUnit - My torture chamber of choice.
* RSpec - Unit Testing for Ruby.
* Selenium - Graphical Unit Testing.
* Test-Driven Development - Way of Life.
Discussion!