

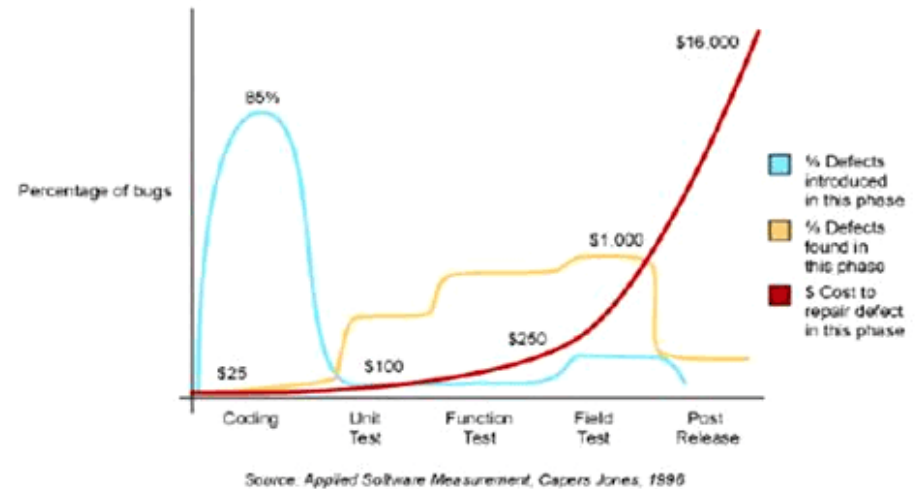
JDojo@Gbg



jsolutions.se

The Problem

- Bugs are costly
- Fixing bugs reported from testers costs even more
- Fixing bugs reported from end users costs even more than "even more"



Source: *Applied Software Measurement*, Capers Jones, 1996

Test-Driven Development



Kent Beck on TDD:

I'm not a great
programmer,
I'm just a good
programmer with
great habits.

IBM & Microsoft Report

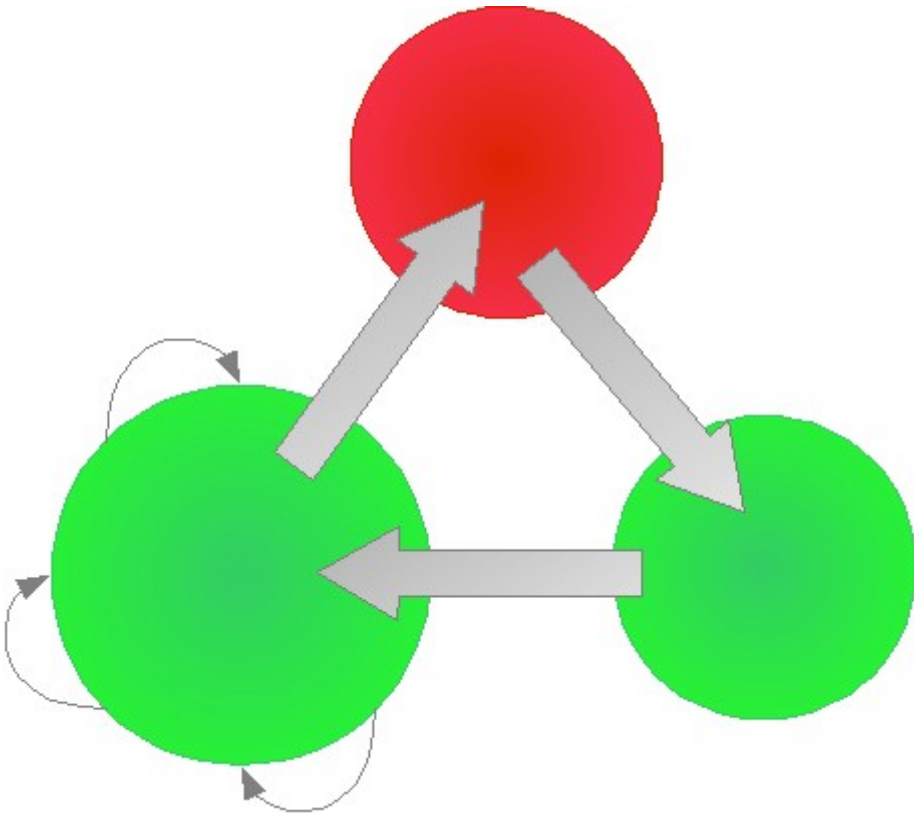


- 40 - 90% fewer bugs discovered during testing
- 15 - 35% more time spent on development

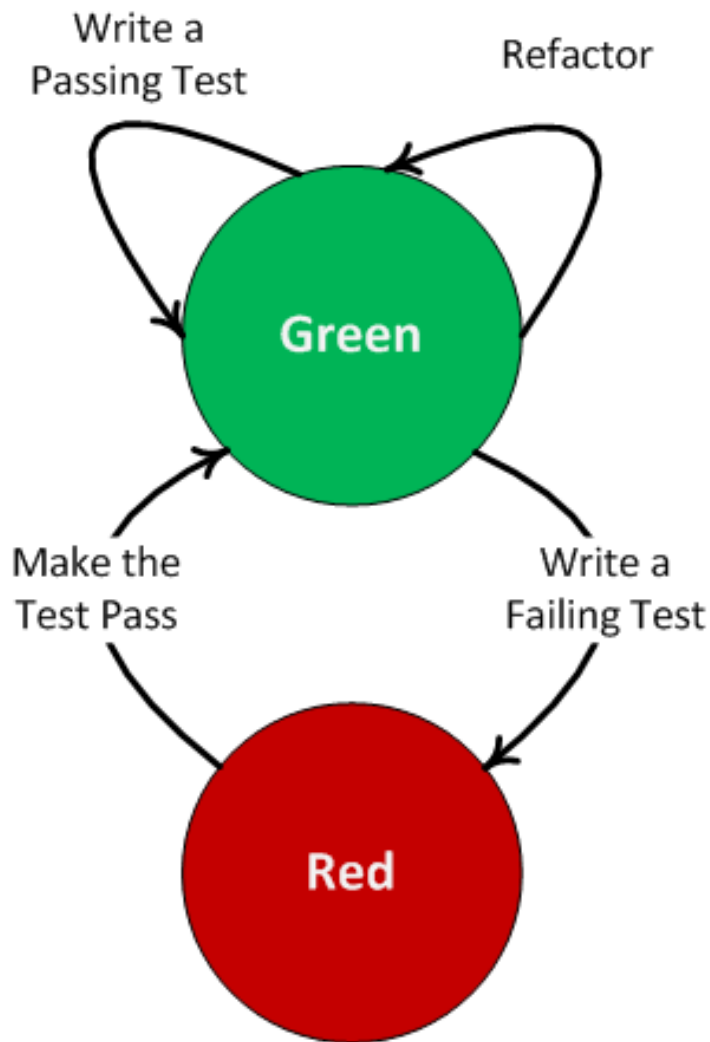


Red, Green, Refactor

- Write a test, get it to compile, get a red bar
- Quickly make the test pass, get a green bar
- Refactor, "clean code", run tests frequently - the "bar" should stay green



Another Take



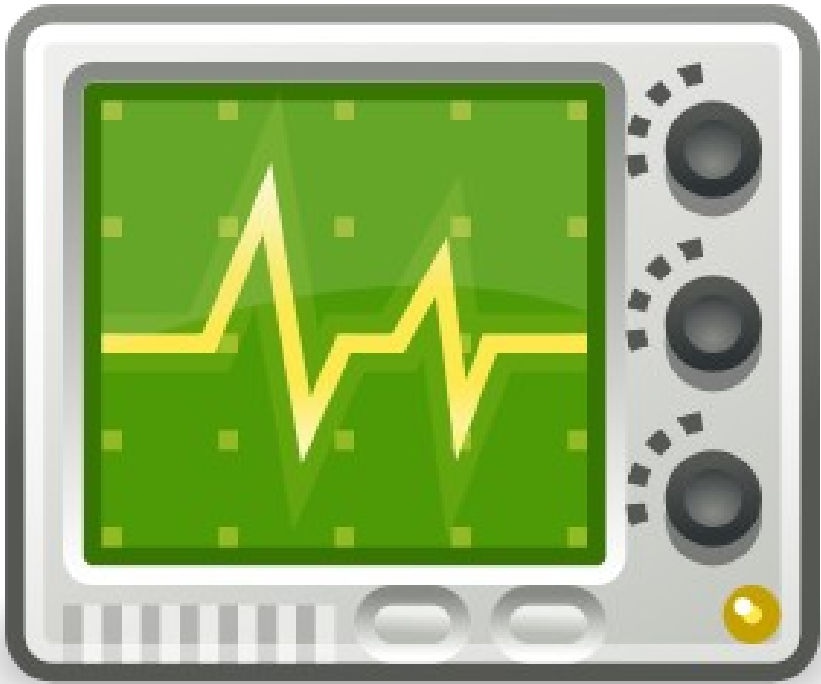
- write test first
 - write smallest possible implementation
 - clean up
- no "dead" code - it's all covered
- nice API - designed outside in

Baby Steps

- as small as possible
- as large as is comfortable
 - experience of TDD
 - technology knowledge
 - boldness
 - understanding of the problem domain



Characteristics of a Good Test

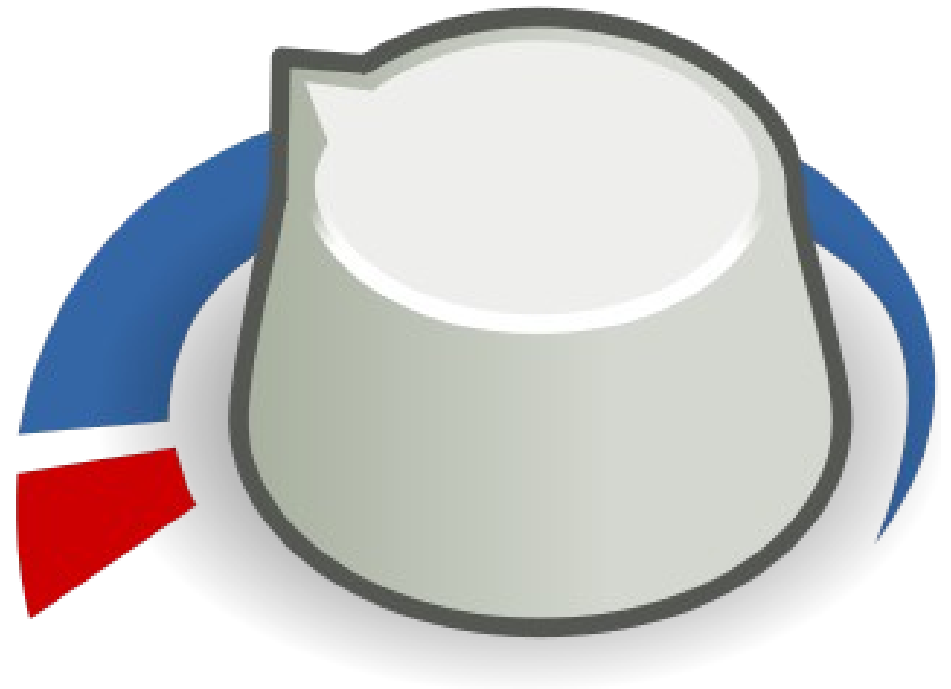


- Communicates how and why a system behaves in a certain way
 - test name
 - comments*
- Fast
- Repeatable

Enough Tests?

Kent Beck's book Test-Driven Development

- test anything that could break
- if code is simple, testing should be simple too
- test both happy and sad paths
- border conditions



Benefits



- More confident
- More productive
- More reliable
- More agile

Drawbacks

- More code to maintain
- Takes longer
- Easy to learn, but harder to master



TDD vs Traditional Testing



Doesn't remove the need for regular, traditional testing:

- Exploratory testing
- QA testing
- User Interface testing
- Performance testing
- ...

JDojo@Gbg



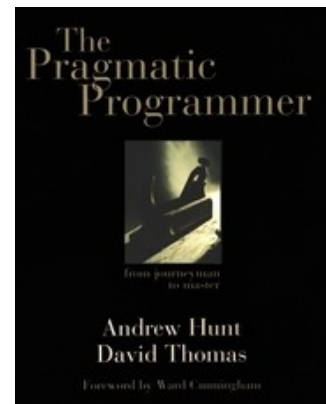
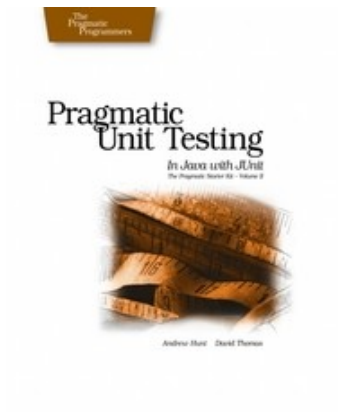
A safe environment to
learn coding skills such
as

Test-driven Development
and writing

clean code

Dave Thomas - Kata

"Pragmatic Dave":
A kata is an exercise in karate where you repeat a form many, many times, making little improvements in each.



The Coder's Dojo

Concept presented at XP 2005 by
Laurent Bossavi and Emmanuel Gaillot



Coding Dojo



- A moderator presents the kata
- Solve the kata, and ...
- Learn from others, teach what you know
- Short retrospective

Kata



- Small programming exercise - 1-2 hours
- Start from scratch
- Strictly test-driven
- Next time, start over and try new approach
- Practice until you master the kata

Dojo Rules

- Don't criticize until we have a green bar
- Strict usage of TDD
- Everyone's obligation to point out any non-TDD behaviour



JDojo@Gbg



Fredrik Wendt
jsolutions.se

More

- <http://codingdojo.org>

Kata Variations

- Prepared kata
- Randori

Behaviour-Driven Development

Given some
pre-conditions,
when something
specific happens,
then some specific
outcome.



String Calculator



Create a method `add(numbers)` where `numbers` is a string.

The method can take 0, 1 or 2 numbers, and will return the sum, for example `"3"` and `"1,2"` would return 3.

For an empty string, return 0.

String Calculator (2)



Allow the add method to handle an unknown amount of numbers.

Allow the add method to handle new lines between numbers (instead of commas):

ok "1\n2,3"

not ok: "1\n,2,3"
(throws Exception)

Kata Calculator (3)



Allow the add method to handle different delimiter:

to change a delimiter, the beginning of the string will contain separate line that looks like this:

```
"/[/delimiter]\n[numbers]"
```

for example `"/;/\n1;2"`

should return 3 where the default delimiter is `","`.

String Calculator (4)



Calling add with a negative number will throw an exception. "negatives not allowed" - and the negative that was passed. If there are multiple negatives, show them all in the exception message.