Refactoring

Mark Reed

Introduction

TDD is defined at a high level as a process of Red-Green-Refactor

To get from Red to Green after faking and triangulating we should write the most obvious implementation

What is the point in risking breaking anything by fiddling with it after that?

It already works

But…

THIS IS HOW WE WOULD GET TO WORK IF WE BELIEVE THE SAYING
"IF IT AIN'T BROKE DON'T FIX IT"
imgflip.com

> ‘If it ain't broke, don't fix it’ is the slogan of the complacent, the arrogant or the scared. It's an excuse for inaction, a call to non-arms.
>
> Colin Powell

Why?

Why should we refactor

A Metaphor

A popular metaphor for refactoring is washing up.

Why Refactor? - Technical Benefits

Readability

Extensibility

Performance

Easier to locate any bugs

Lower cost of change

Lower total cost of ownership

Why Refactor? - Personal Benefits

It is highly satisfying – like a video game in itself

Allows you to take professional pride in what you are doing

You may be the guy needing to understand the code or extend it in the future

Greater sense of job satisfaction

How?

How should we refactor

How to refactor?

Aggressively

Focus on readability first

The IDE is your friend (Rename, Extract, Inline, Remove etc.)

Stay green, use parallel change

Keep going until you are happy that you can refactor no more

Don’t be scared to revisit existing code and refactor that if you spot anything that you think could be improved (as long as it is under test)

When?

When should we refactor

When to refactor?

Constantly

You are green

You break the rule of 3

You are breaking object calisthenics rules

You spot a code smell

You have a domain insight

Summary

Refactoring is an essential part of evolving a good software design.

It is not a vanity project for developers to indulge themselves in some irrelevant optimization just to feel pleased with themselves.

Without it, we can have a working solution,

but it won’t be easily maintainable

or extensible

and that will make anyone who has to work on it:

slower,

more error prone

and ultimately miserable.

Questions?

Thank you

Mark Reed

mark.reed@fdbhealth.co.uk