

# Coding Dojo Challenge XP2013

Emily Bache

[emily@bacheconsulting.com](mailto:emily@bacheconsulting.com)

@emilybache on twitter



# The Coding Dojo

- The dojo is the place where you go every week to practice and learn karate



# Practical Coding Skills

- using IDE keyboard shortcuts
- Pair Programming
- Test Driven Development
- Refactoring
- Designing good Test Cases
- Working incrementally, committing code often
- Designing using SOLID principles
- Object Oriented Paradigm
- Functional Programming



# Learn on-the-job



*Attribution: flickr user Lisamarie Babik*

## Pair Program & Get Stuff Done

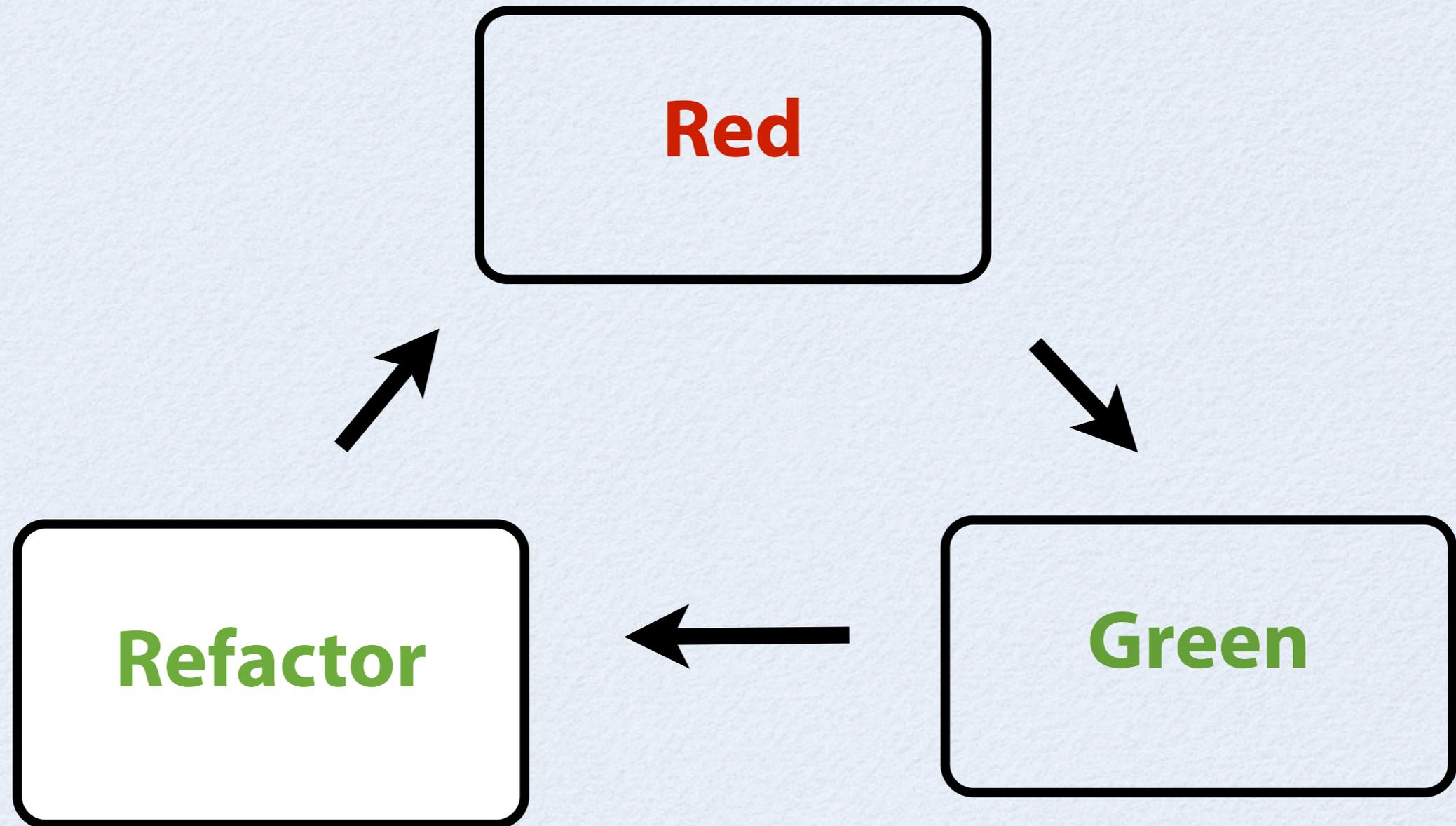
# Learning to Ski



*flickr user fiction300*

*Snowplow vs Parallel Turns*

# Test Driven Development



# Incidental vs Deliberate

## Incidental Practice:

Repeatedly doing something you can already do

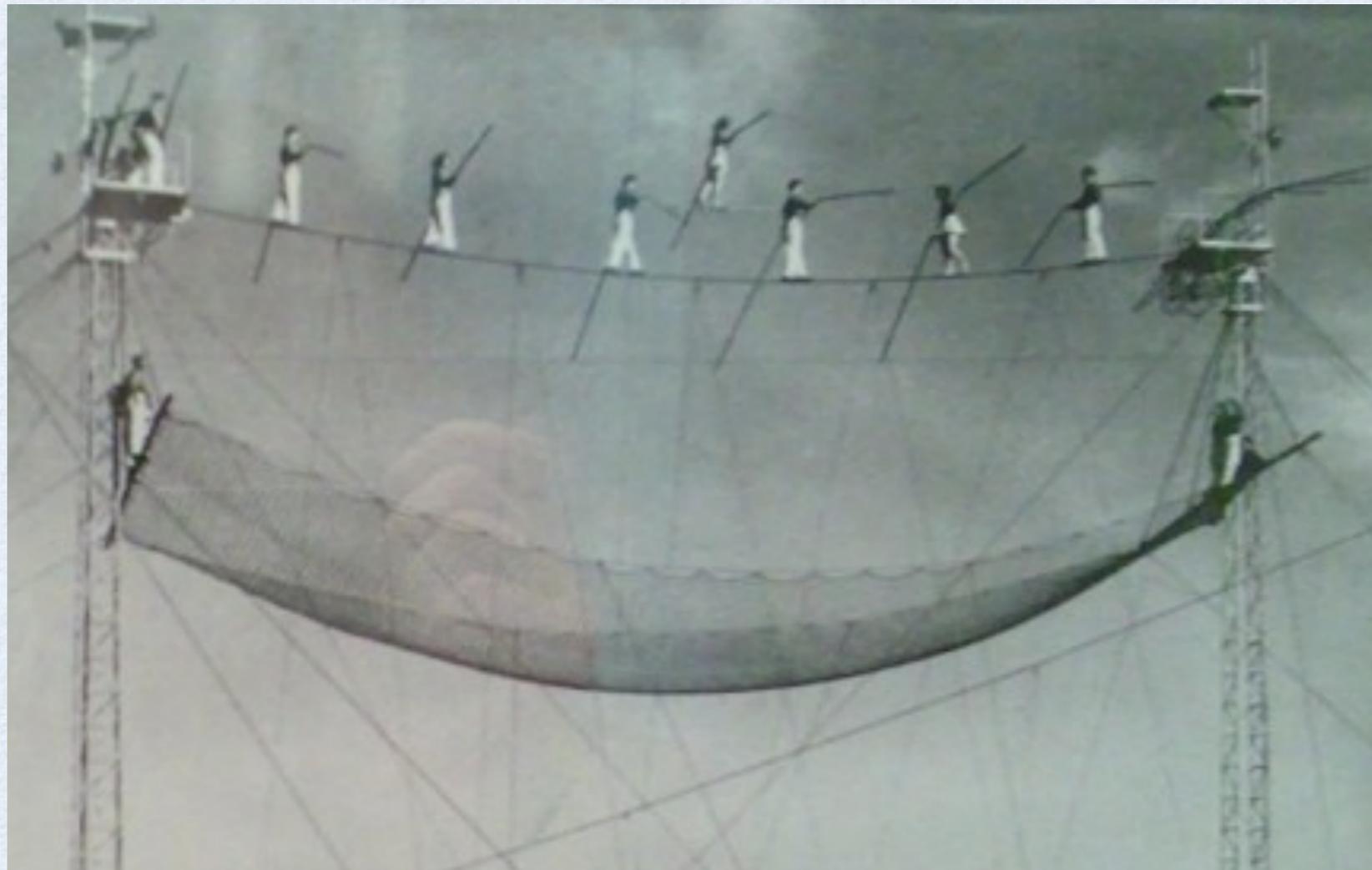
## Deliberate Practice:

Trying to do something you can't comfortably do

Breaking down a skill into components you practice separately

# Deliberate Practice

- Need to feel safe
- Need to feel motivated



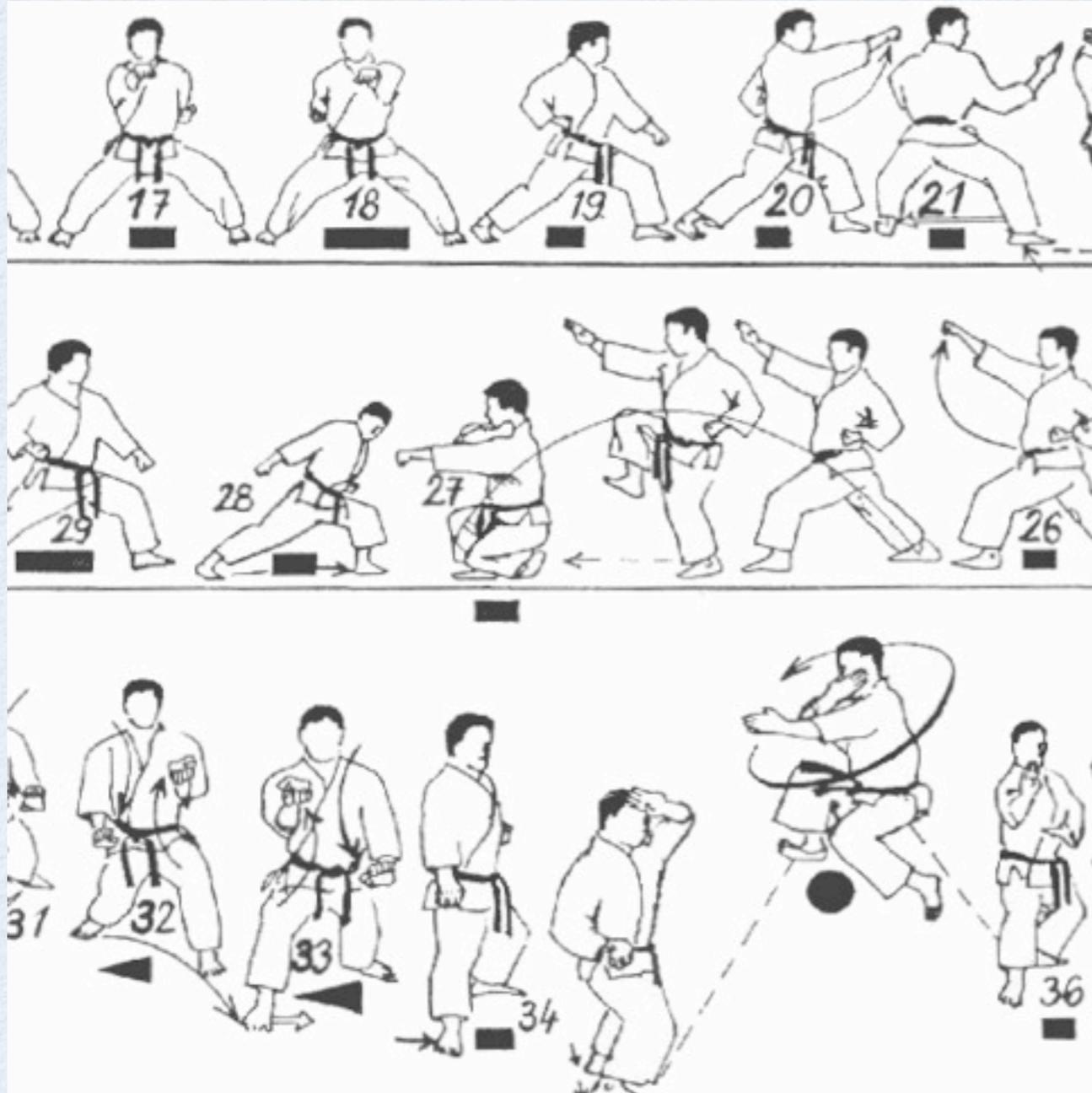
# Training

“Under pressure, you don’t rise to the occasion, you **sink to the level of your training.**”

That’s why we train so hard”

*-- A US Navy Seal  
(according to twitter)*

# Code Kata: TDD moves



‘Pragmatic’ Dave Thomas

# Coding Dojo Summary

- Better coding skills
- Deliberate Practice
- A safe environment
- Discuss actual code
- have fun!



# THE CODING DOJO HANDBOOK

Available from:  
<http://leanpub.com>

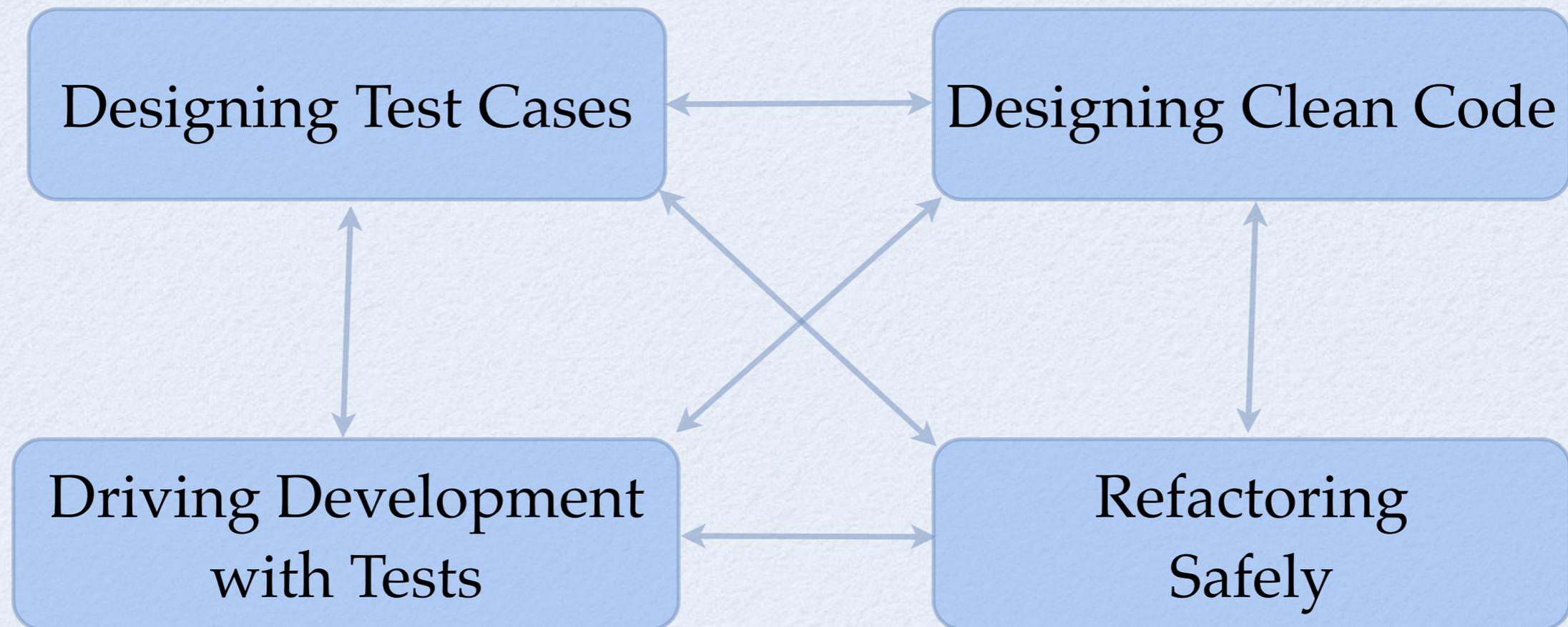
*a practical guide to  
creating a space  
where **good** programmers  
can become **great** programmers*



Emily Bache

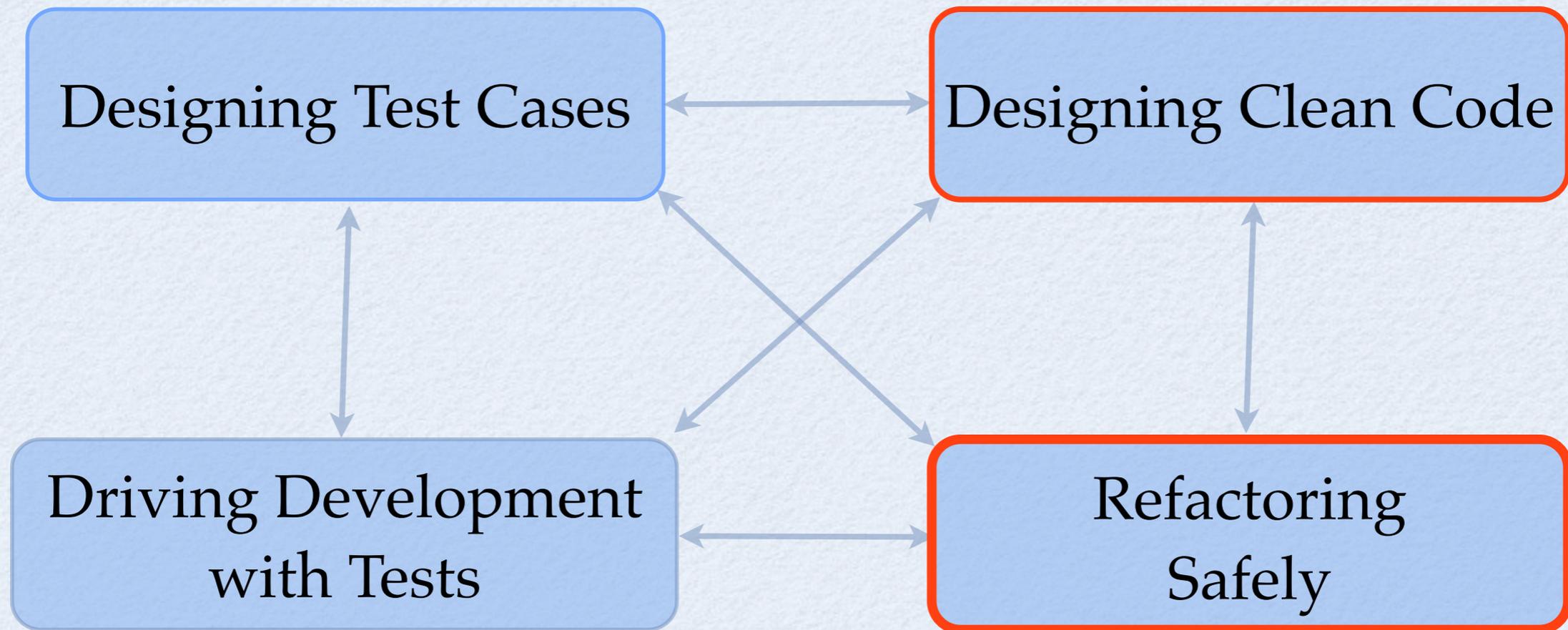
Foreword by Robert C. Martin

# TDD skills



*In the dojo we can focus on one at a time*

# Tennis Kata



# Tennis

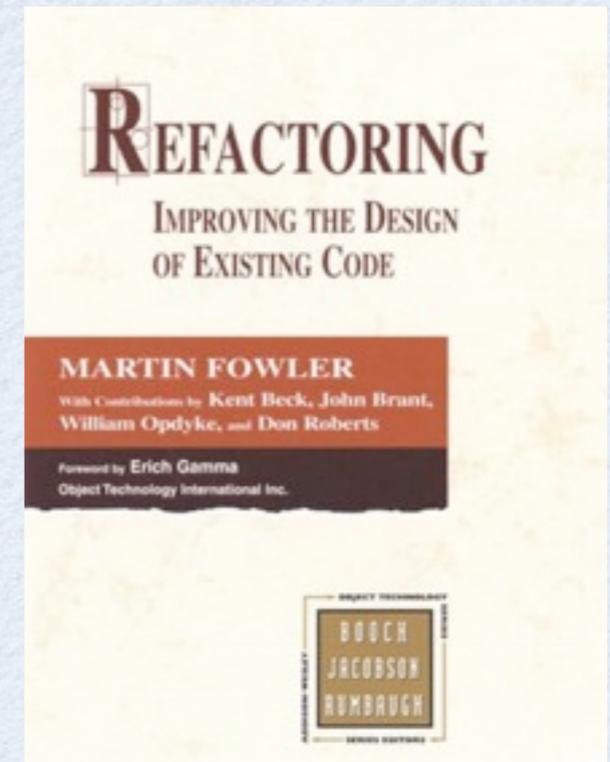
- Your colleague wrote this code for the Tennis Society.
- “Tidy it up and give him feedback” says your boss



# Refactoring

***Refactoring*** (noun): a change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behaviour.

*(Martin Fowler, Refactoring p53)*



# Extract Method

```
51 ByteArrayOutputStream byteStream = new ByteArrayOutputStream()
52 GZIPOutputStream zipOut = new GZIPOutputStream(byteStream)
53 OutputStreamWriter tempOut = new OutputStreamWriter(zipOut)
54
55 // Compress original output and put it into byte array.
56 tempOut.write(new String(responseChars));
57 tempOut.write("Content-Length: " + byteStream.size() + "\n");
58 // Gzip stream explicitly closed.
59 zipOut.close();
60 tempOut.close();
61
62 // Update response headers.
63 res.setHeader("Content-Encoding", "gzip");
64
65 // Set response output stream.
66 res.getOutputStream().write(byteStream.toByteArray());
67
68 // System.out.println("Response size: " + byteStream.size());
69
70 }
71 }
72
```

The screenshot shows an IDE interface. The main window displays a Java code snippet. A context menu is open over the code, with the 'Refactor' option selected. A sub-menu is open from 'Refactor', and the 'Extract Method...' option is highlighted. Below the code editor, there is a 'Tasks' panel with the following content:

| C | ! | Description  |
|---|---|--|
|   |   | The import millhouse.keytopic.tools.codeparser.KCode |

# Extract Method

- ❑ Create a new method, name by intention
- ❑ Copy extracted code from source method to new target method
- ❑ Scan for local variables.
- ❑ Temporary variables local to method?
- ❑ Local-scope variables modified? Return changes back to parent method.
- ❑ Pass local-scope variables as parameters.
- ❑ Compile.
- ❑ Replace extracted code in source method with call to target method.
- ❑ Compile and test.

# Take small steps

- Stay close to green
- Follow refactoring steps
- One small change at a time
- Don't rewrite it from scratch



*flickr user San Diego Shooter*

# Lean on the Tests



*flickr user ell brown*

- For this kata - the tests are pretty comprehensive
- Run tests often!

# Cyberdojos

<http://cyber-dojo.com>

- Python - 328D36
- Ruby - 9800B7
- Java - 21BC8B
- C++ - C05A79
- C# - 5F612E

# Coding Dojo Challenge

Emily Bache

[emily@bacheconsulting.com](mailto:emily@bacheconsulting.com)

@emilybache on twitter

