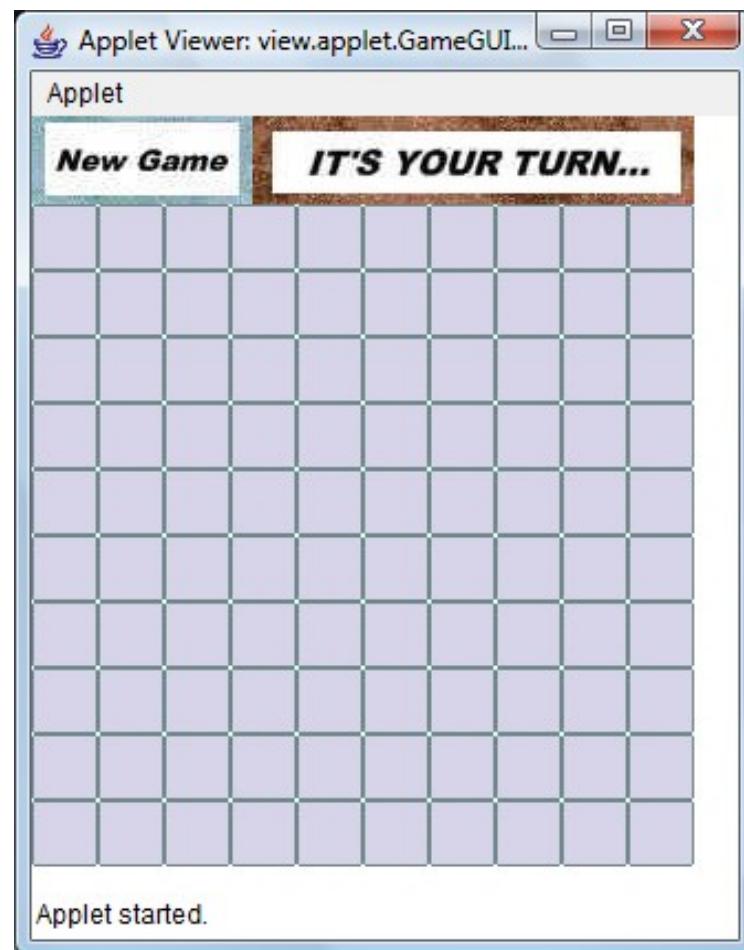
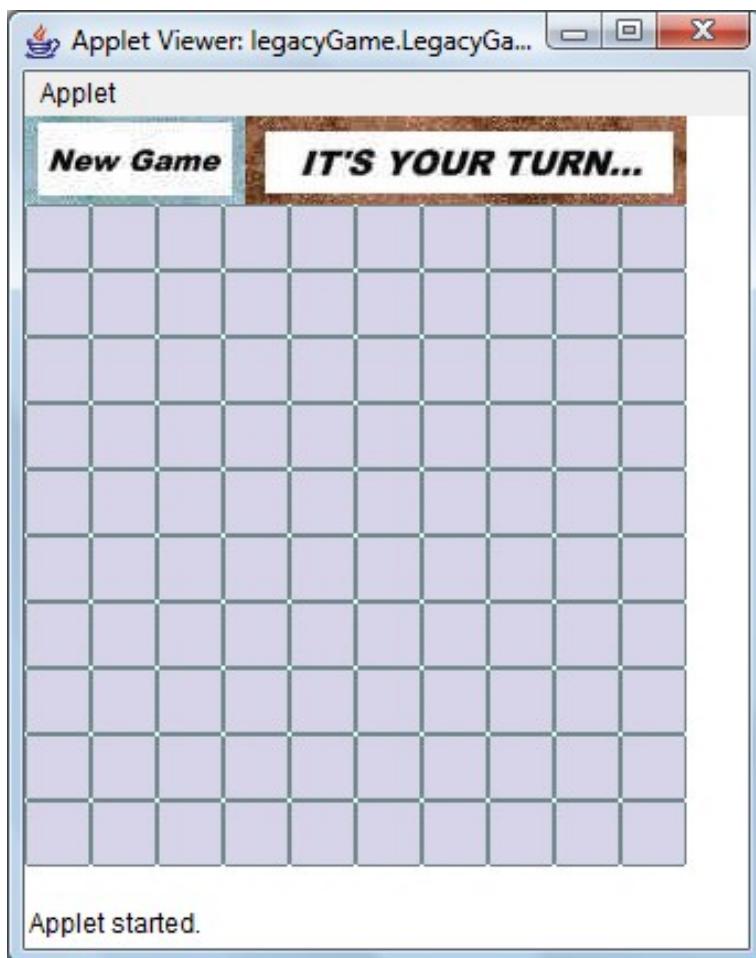


Clean Code

Emily Bache
IBS JavaSolutions
www.jsolutions.se



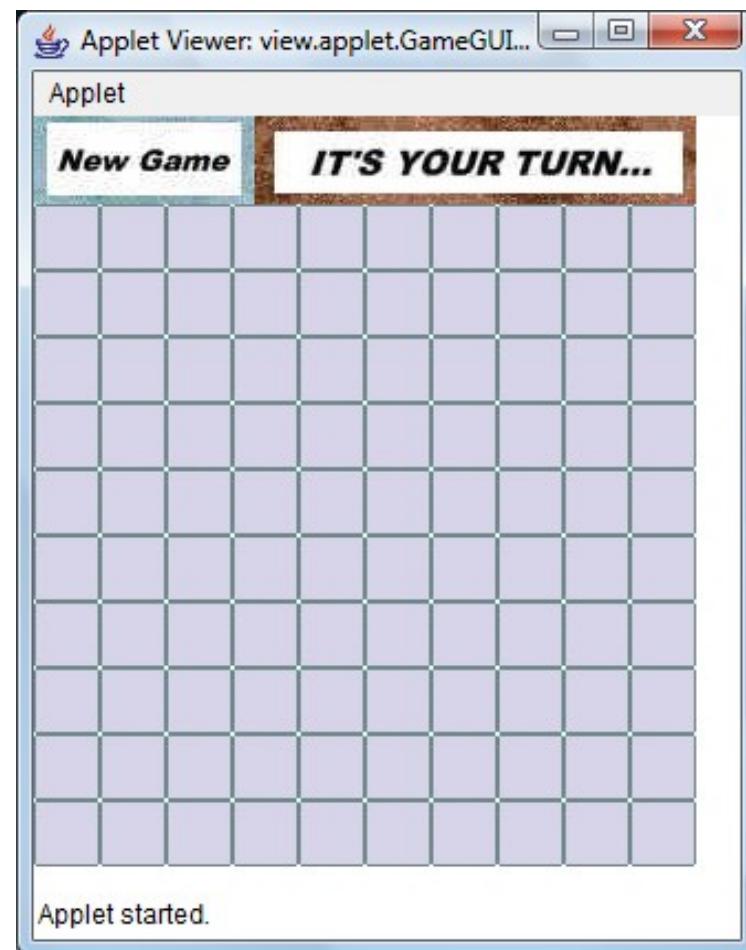
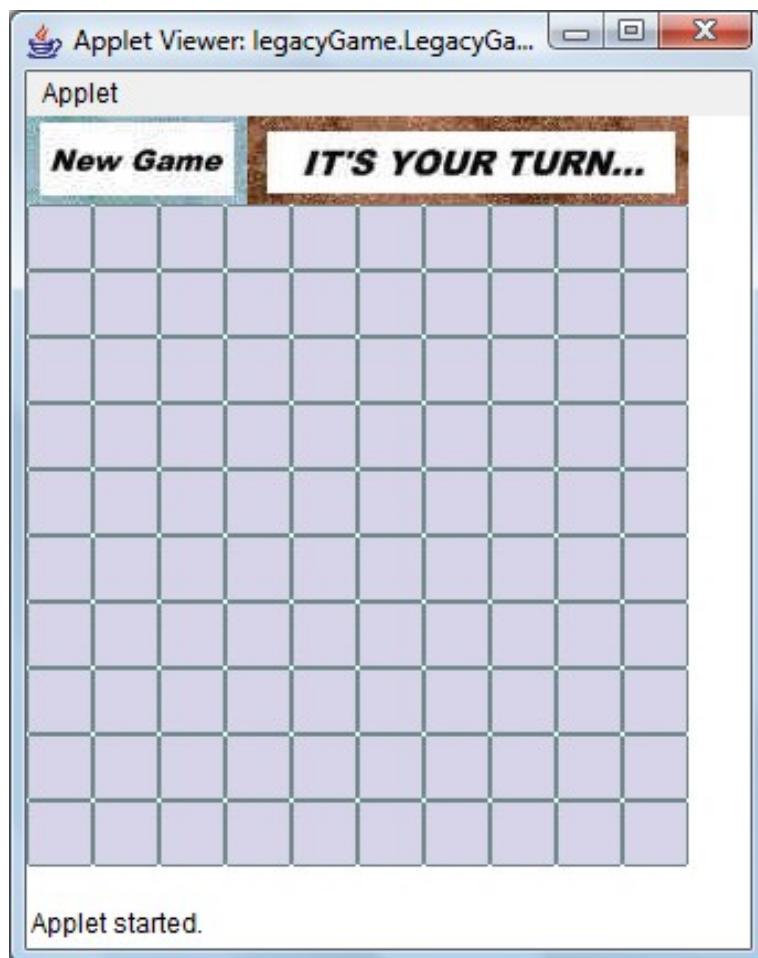
Spot the Difference



A tale of two applets...

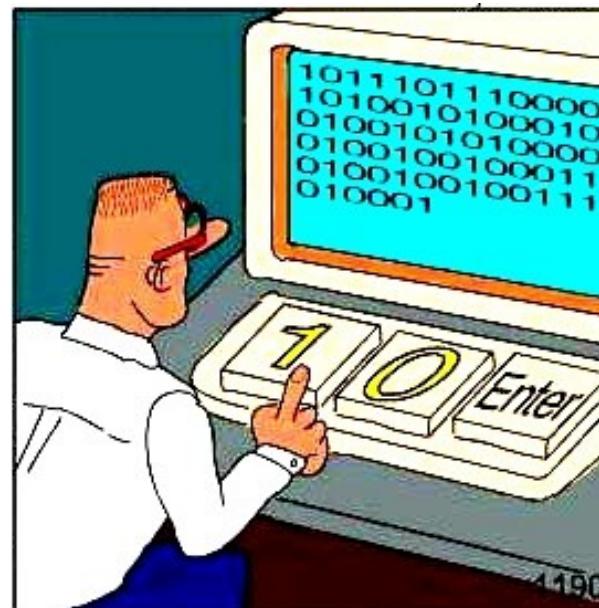
- One enormous class
1300 lines
- 26 data members
- 40 public methods
“setFlagsForLaterProcessing”
- Long functions
- Badly named variables
- Tens of classes
- Several packages
- “Board”, “Game”,
“Strategy” classes
- Small methods
- Comprehensive unit tests

The User can't tell



Clean code == happy programmers

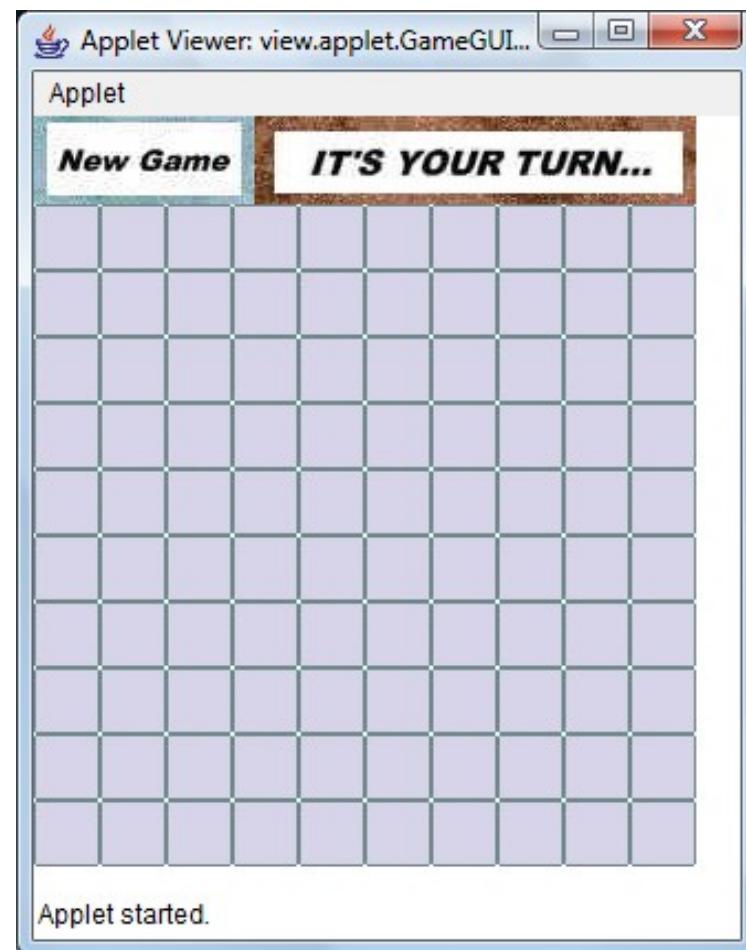
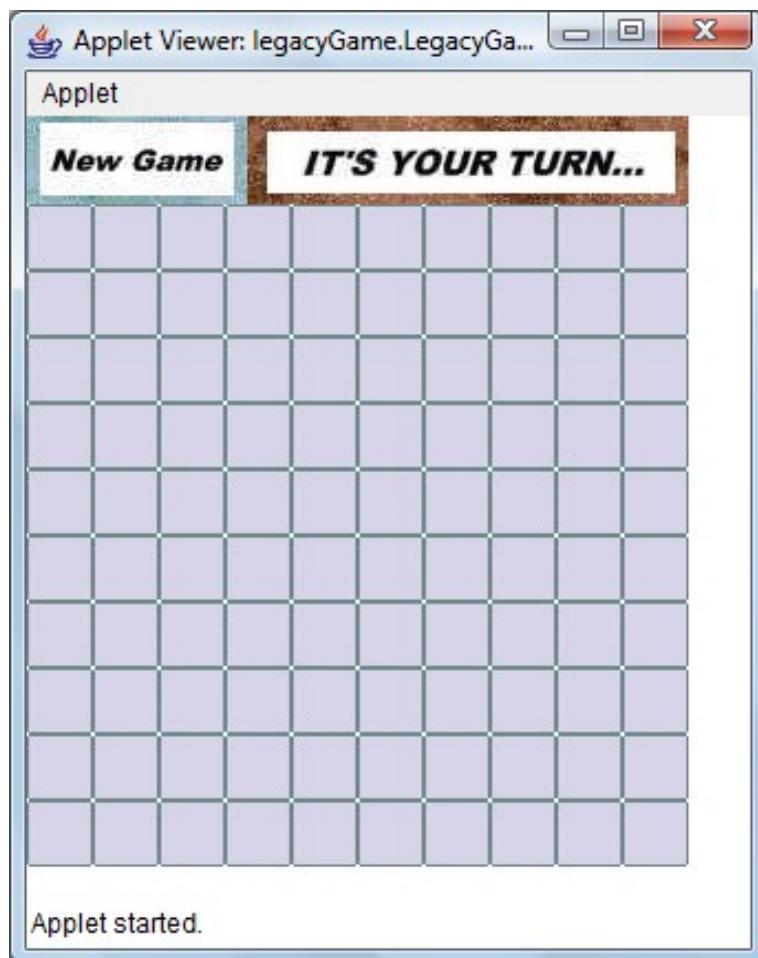
Horrible code sucks the life out of you



REAL Programmers code in BINARY.

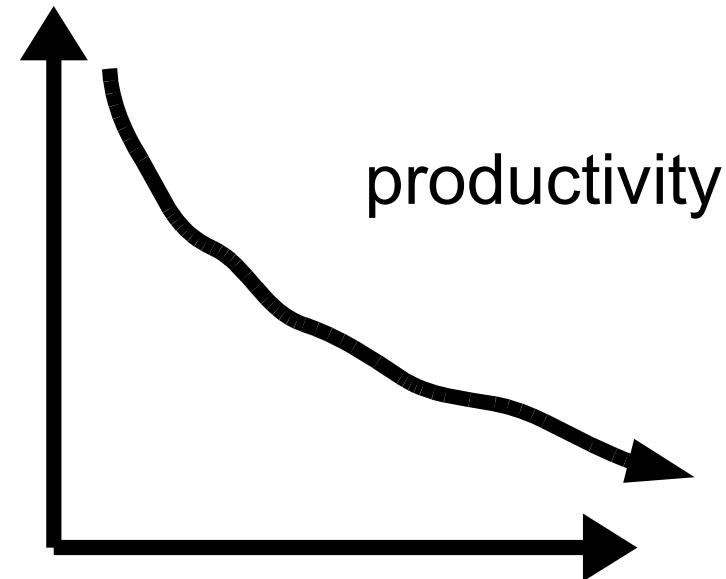
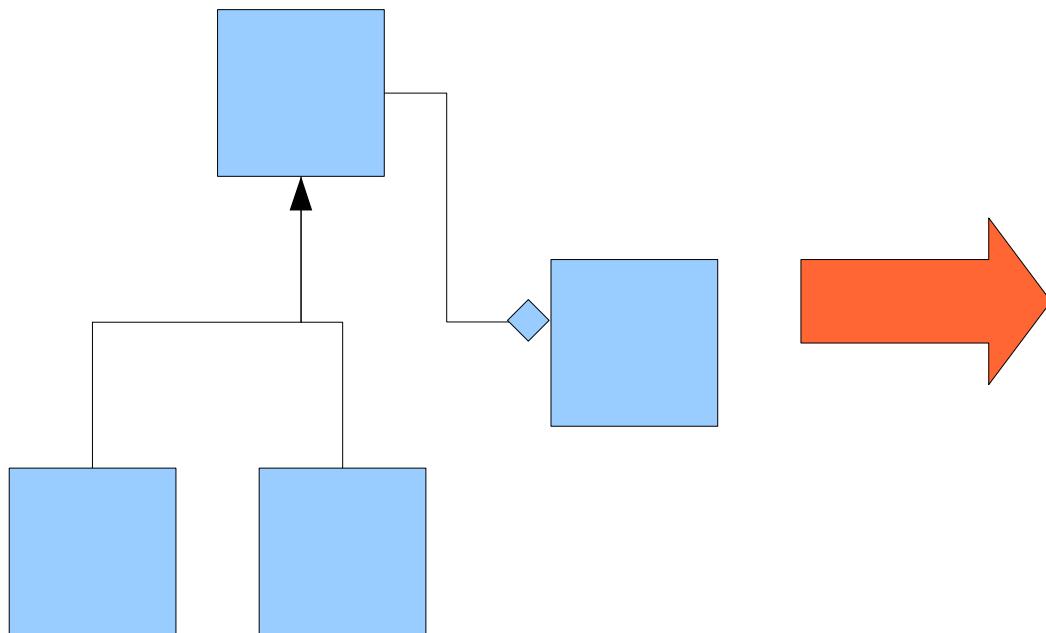
(Terrible tools do too)

Can your manager tell?

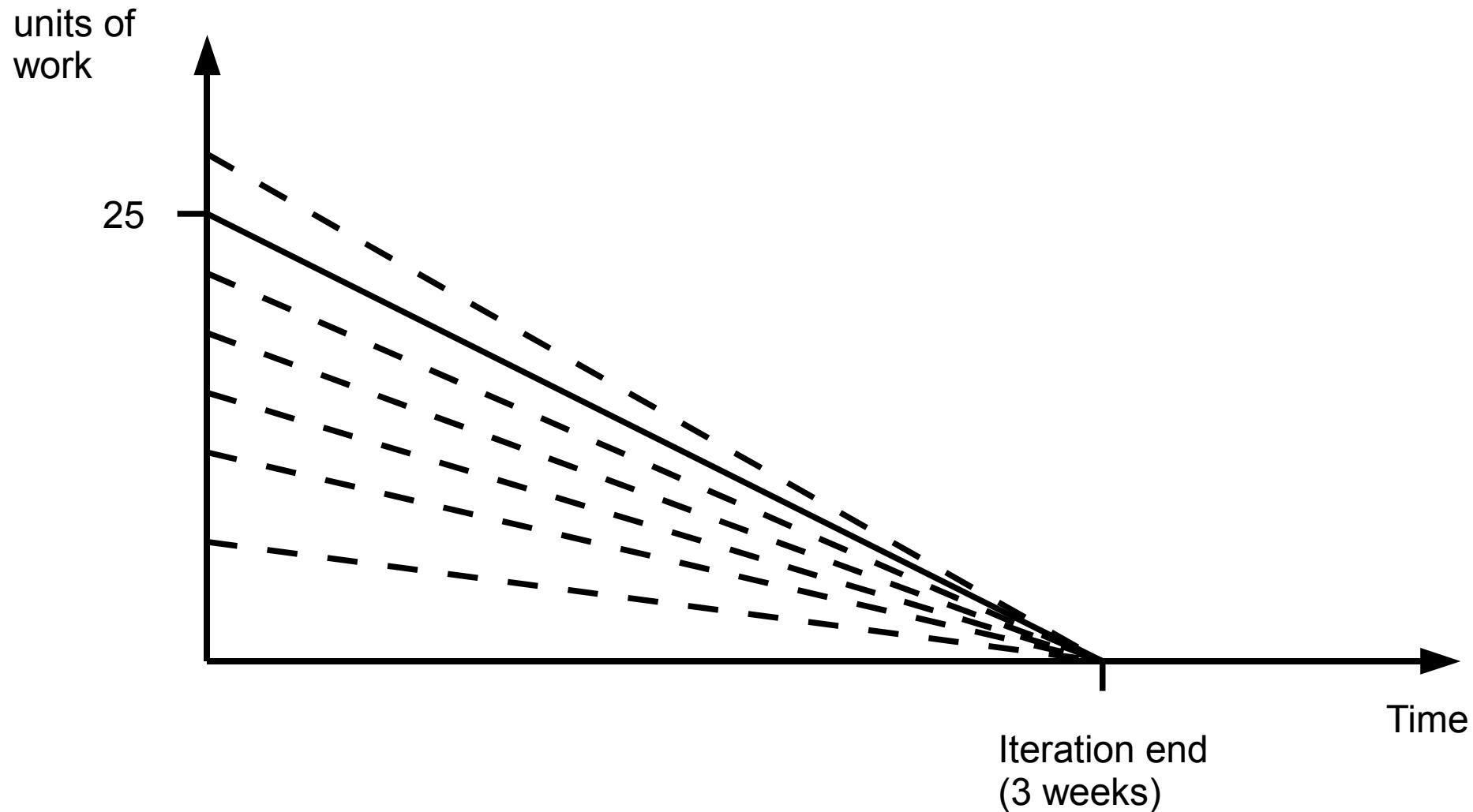


Productivity trend

- As mess builds, productivity decreases.



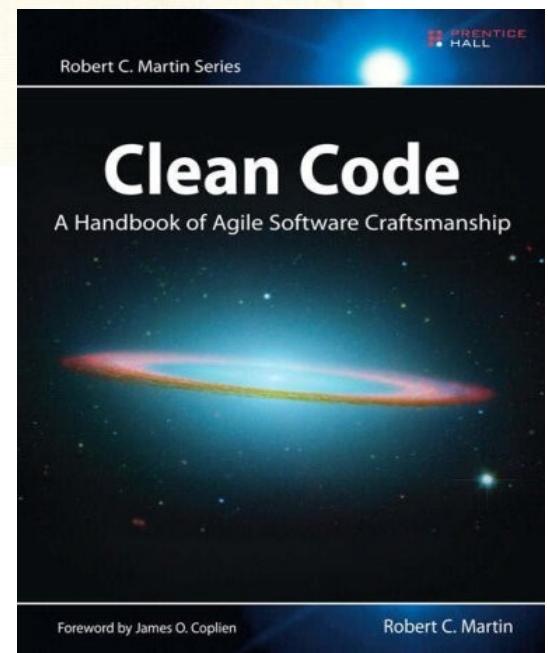
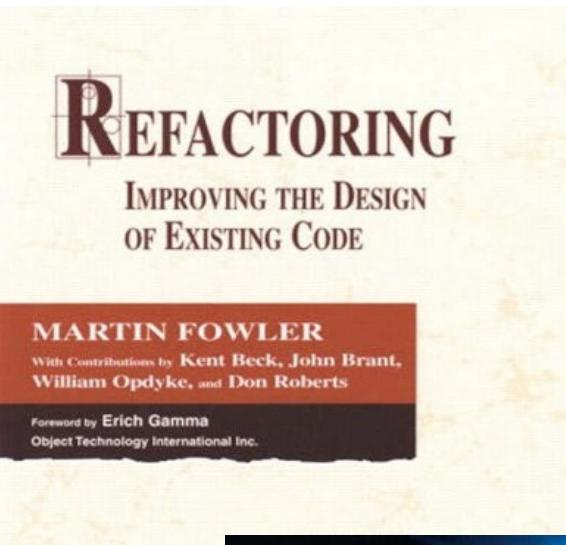
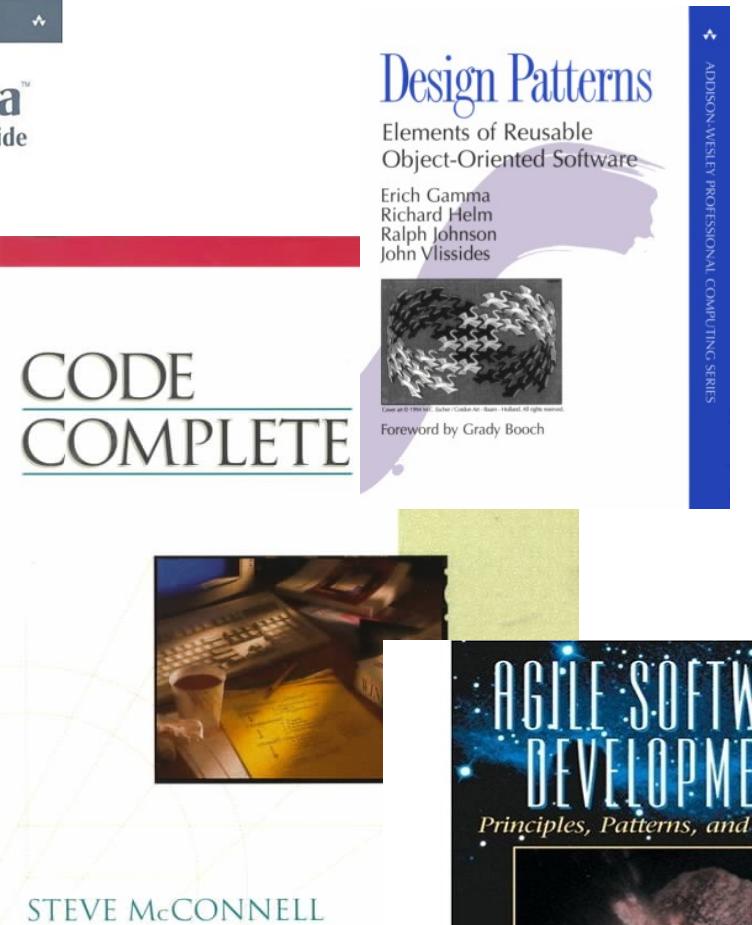
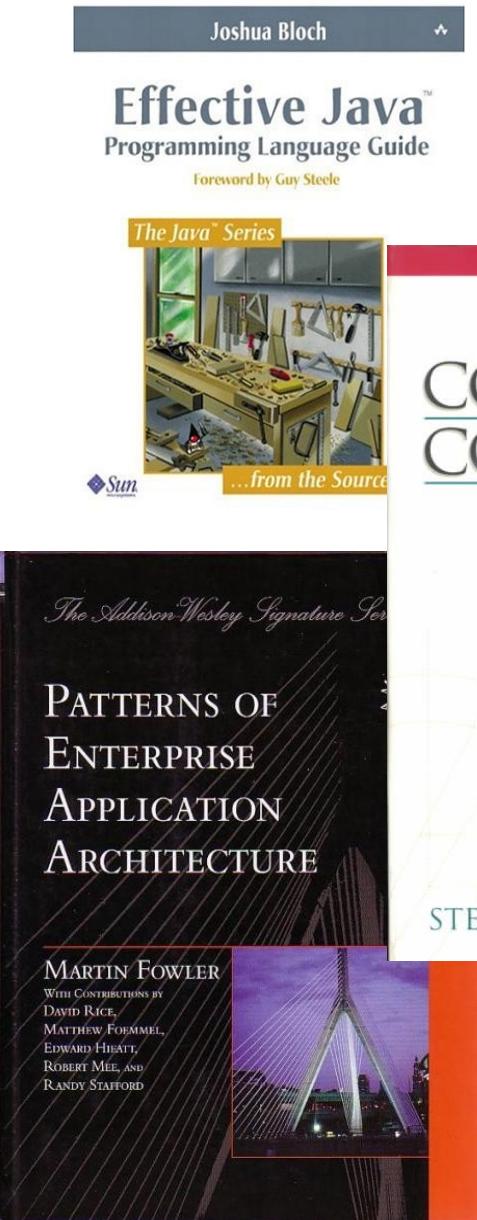
How the productivity slide happens



Coding is like Cooking



Design matters



www.jsolutions.se

Ward Cunningham

“You know you are working on clean code when each routine you read turns out to be pretty much what you expected.”

“You can call it beautiful code when the code also makes it look like the language was made for the problem”



*Ward Cunningham, quoted on p11 of “Clean Code” by Bob Martin
JavaSolutions.se*

Kent Beck

Rules for simple design:

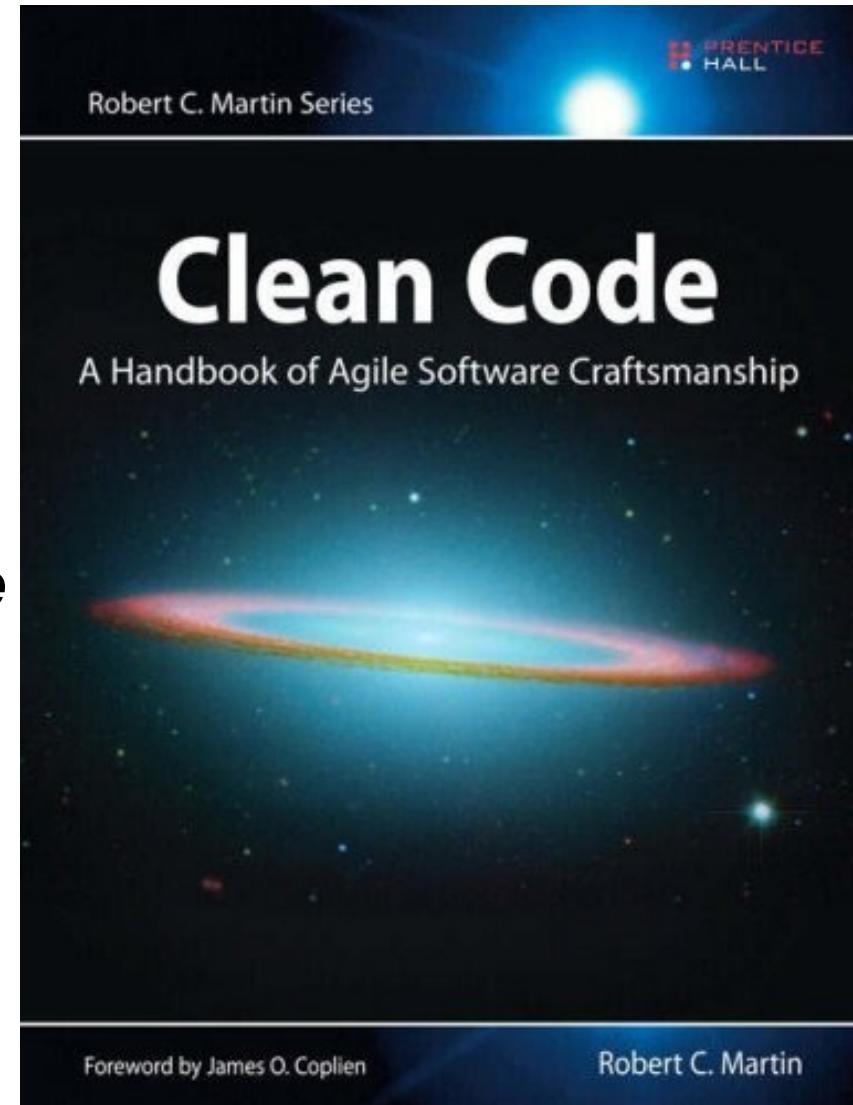
1. Appropriate for the intended audience
2. Every idea communicated
3. Factored. (no duplication)
4. Minimal. (fewest elements)



P109, XP explained 2nd ed., paraphrased slightly

Object Mentor school of clean code

- Uses standard language specific idiom & conventions
- Very few comments
- Small classes & methods
- Everything in one function at *same level of abstraction*
- Reads from top to bottom, more detail at each stage
- ... and much much more



Clean code: Refactoring and Tests

The secret of clean code:

- first write dirty code, *then clean it*

Agile manifesto

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

Code with automated tests over well designed code

I am stuck with a big ball of mud



A Car and an Architect



The Rewrite

How to deal with the big ball of mud?

Rewrite it! All of it!



The Grand redesign in the sky

The Grand redesign project must:

- Implement all old functionality
- Probably several old bugs too
- Probably without full documentation of old system
- Probably learn new technology
- Keep up with new features and bug fixes
- Avoid having their project cancelled in next budget
- Write **cleaner code** than last time!



When you're sinking into a big ball of mud:

Step 1: **Stop Digging!**

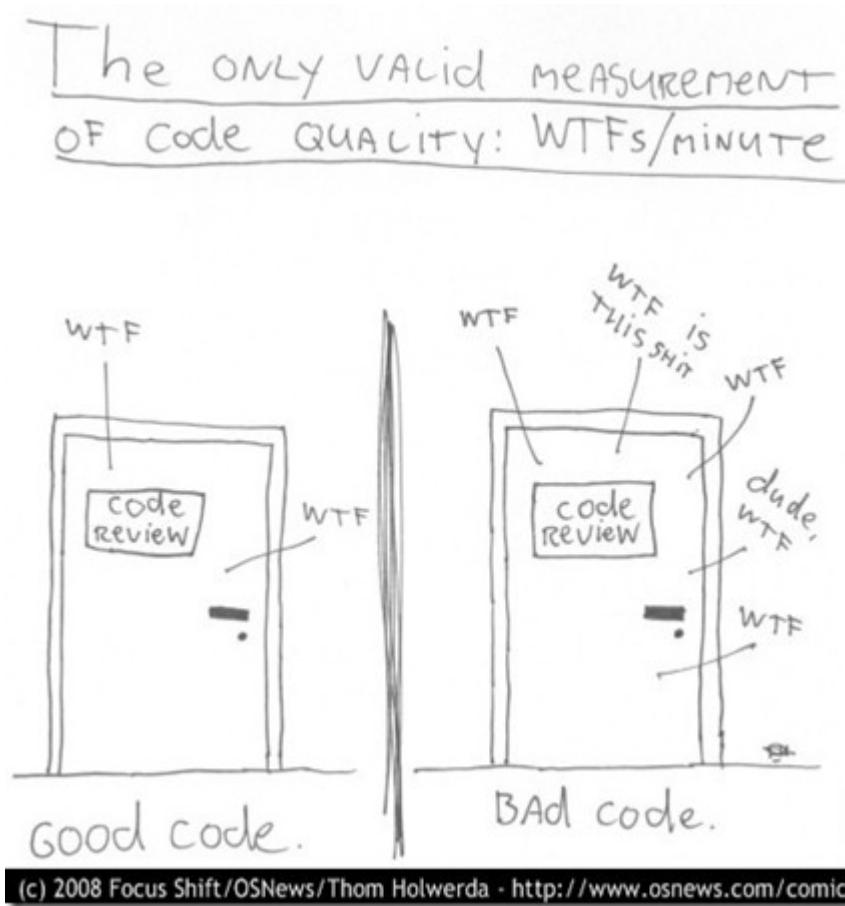


Leave it cleaner

- Every time you change code:
- Improve a variable name
- Extract a method
- Write a test

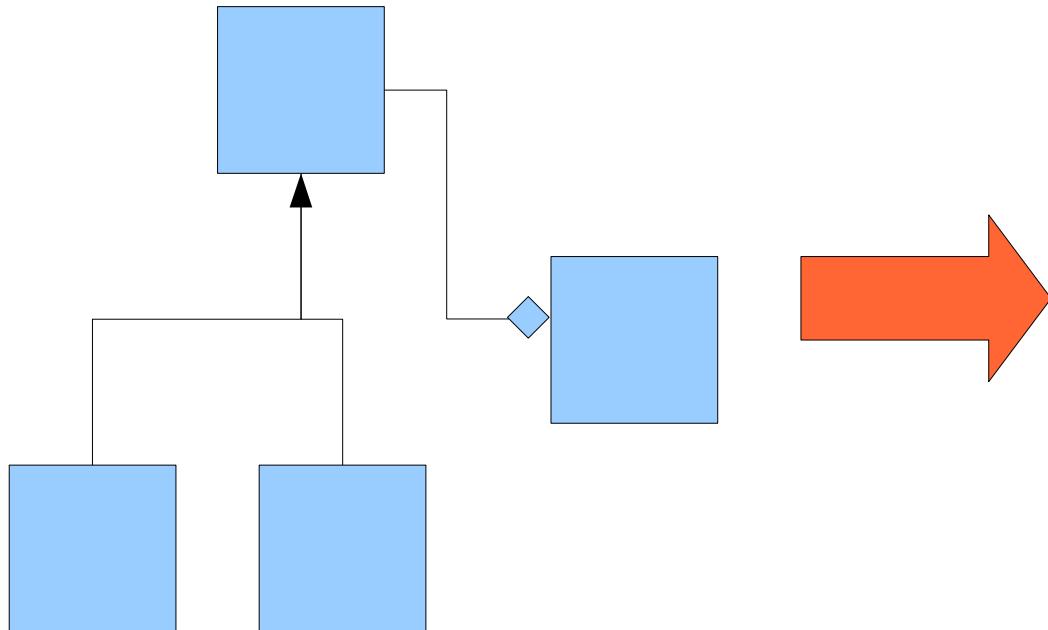
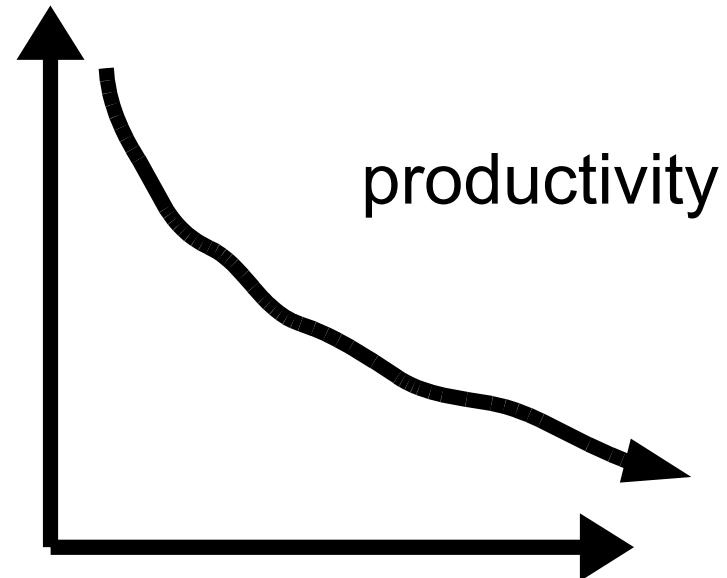


Put code quality on the agenda



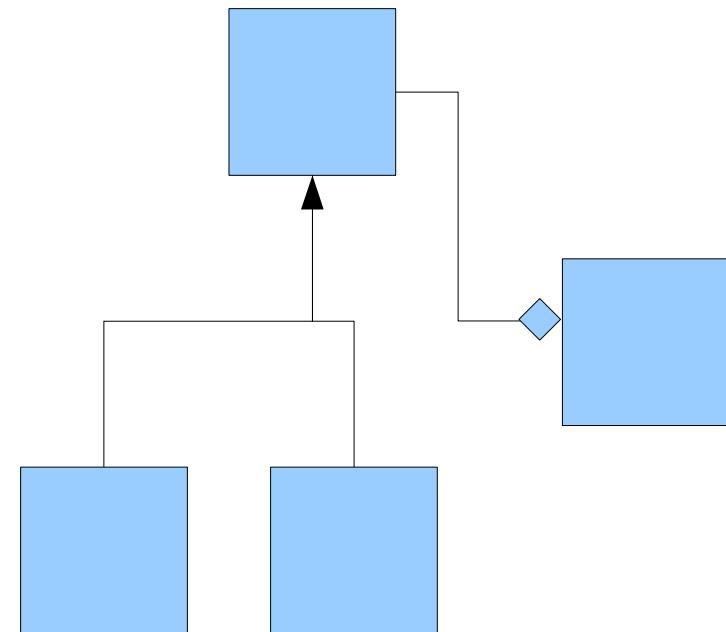
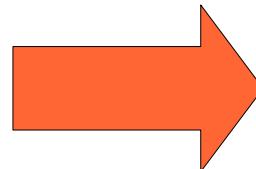
Productivity trend

- As mess builds, productivity decreases.

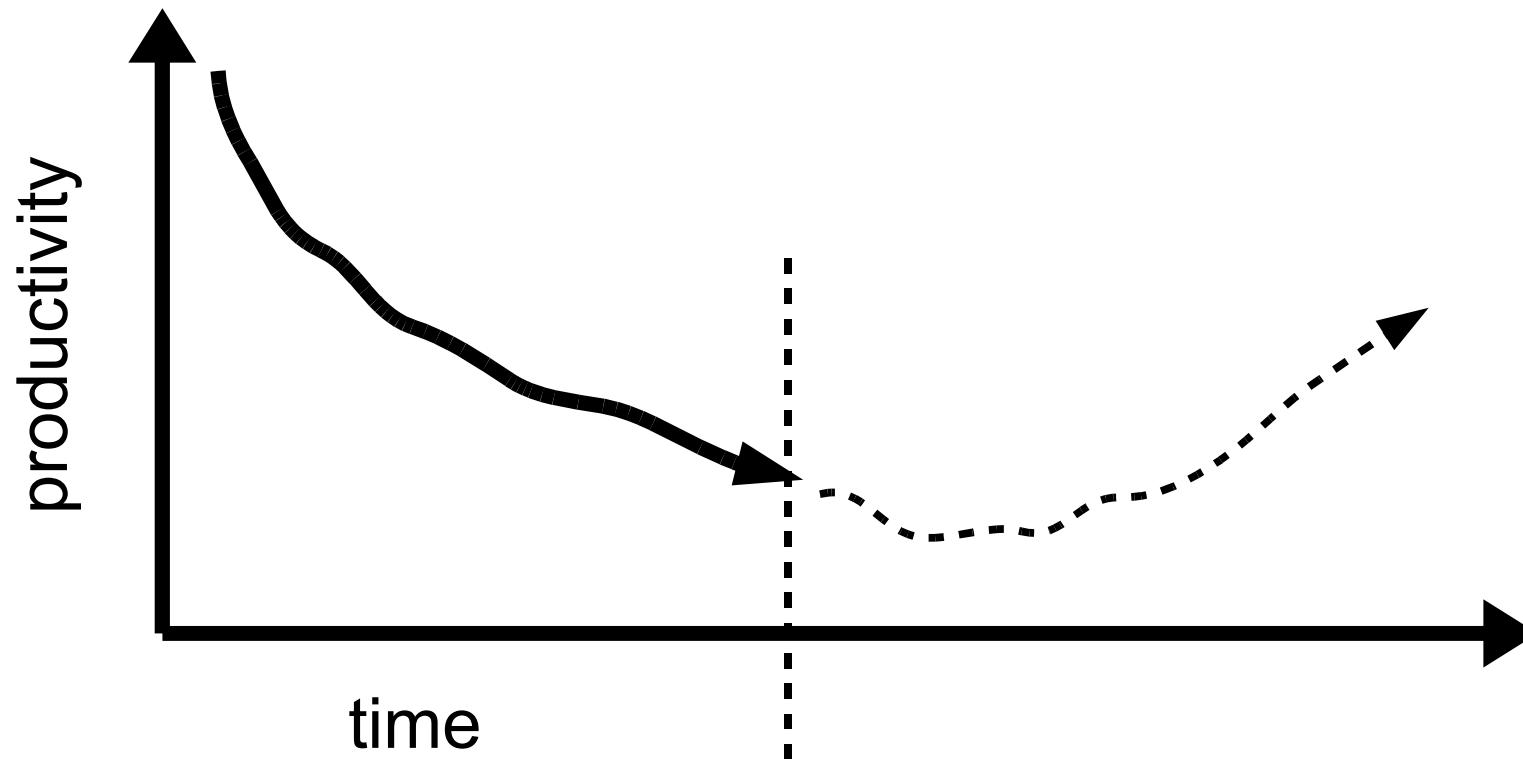


A productivity intervention

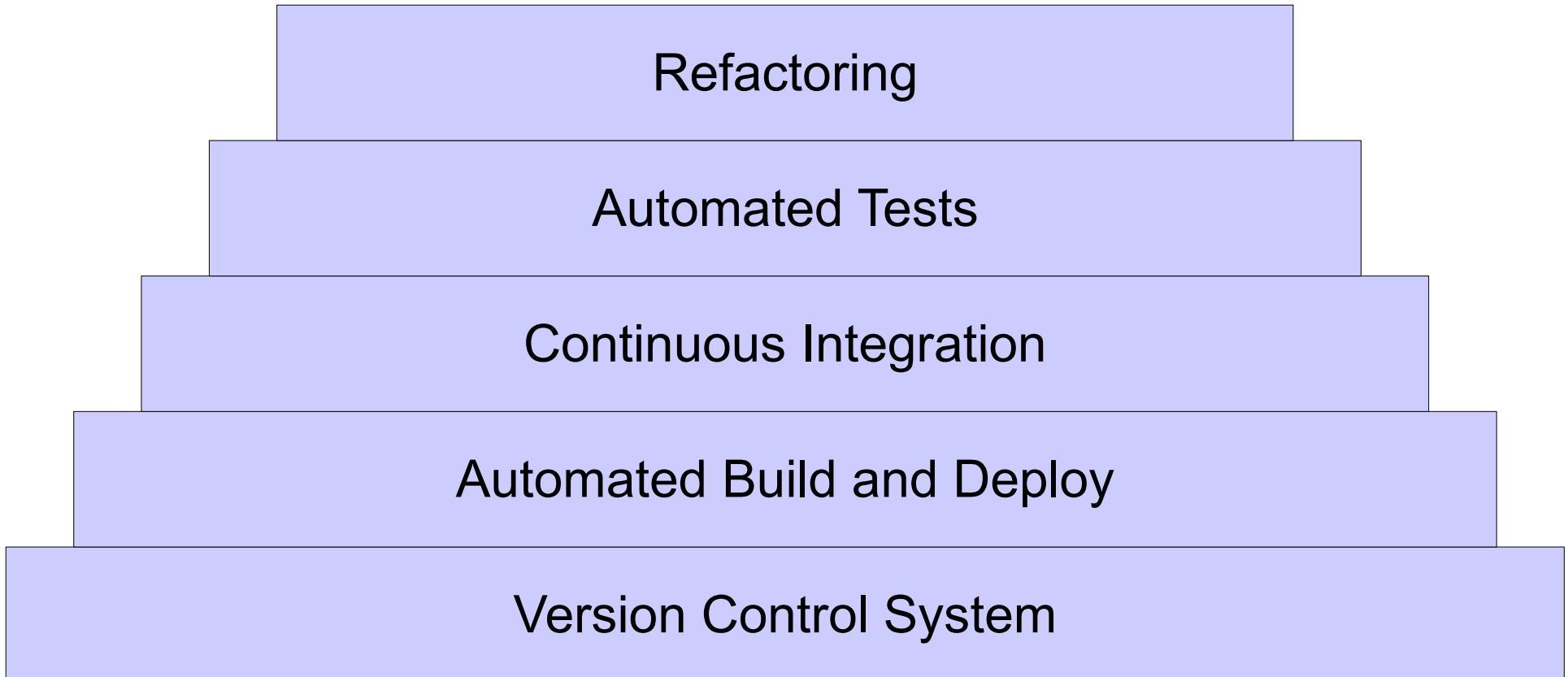
- The whole team should agree change is needed
- You'll need to invest some time and money



Productivity turnaround

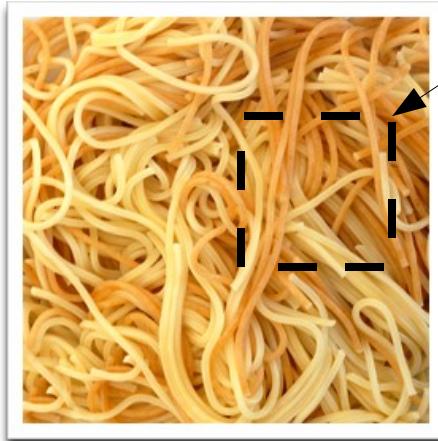


The Productivity Practices Stack

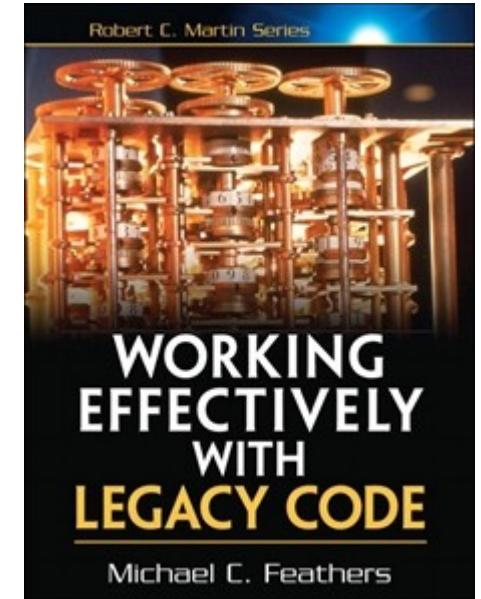
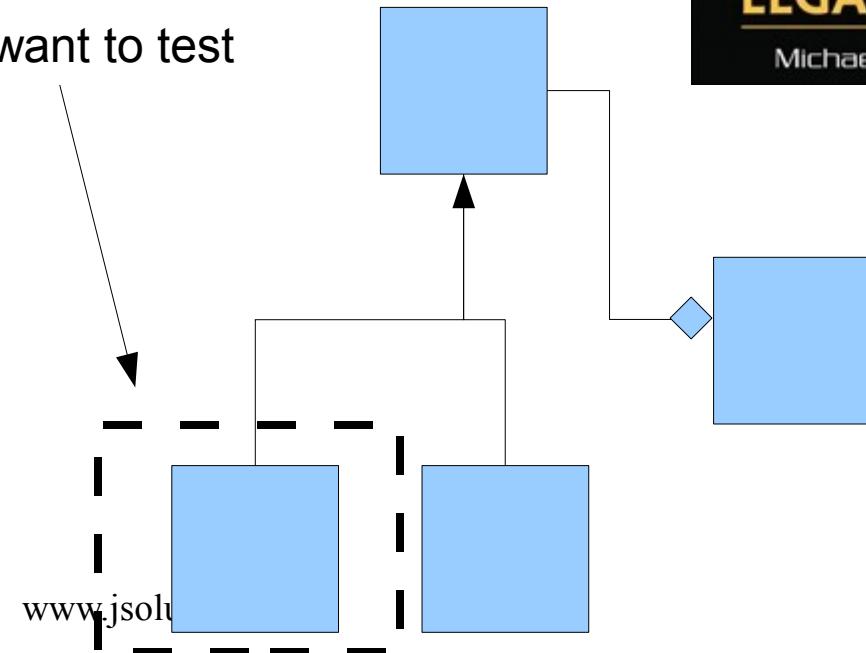


Tests and Legacy Code

- How to unit test spaghetti?



The unit I want to test

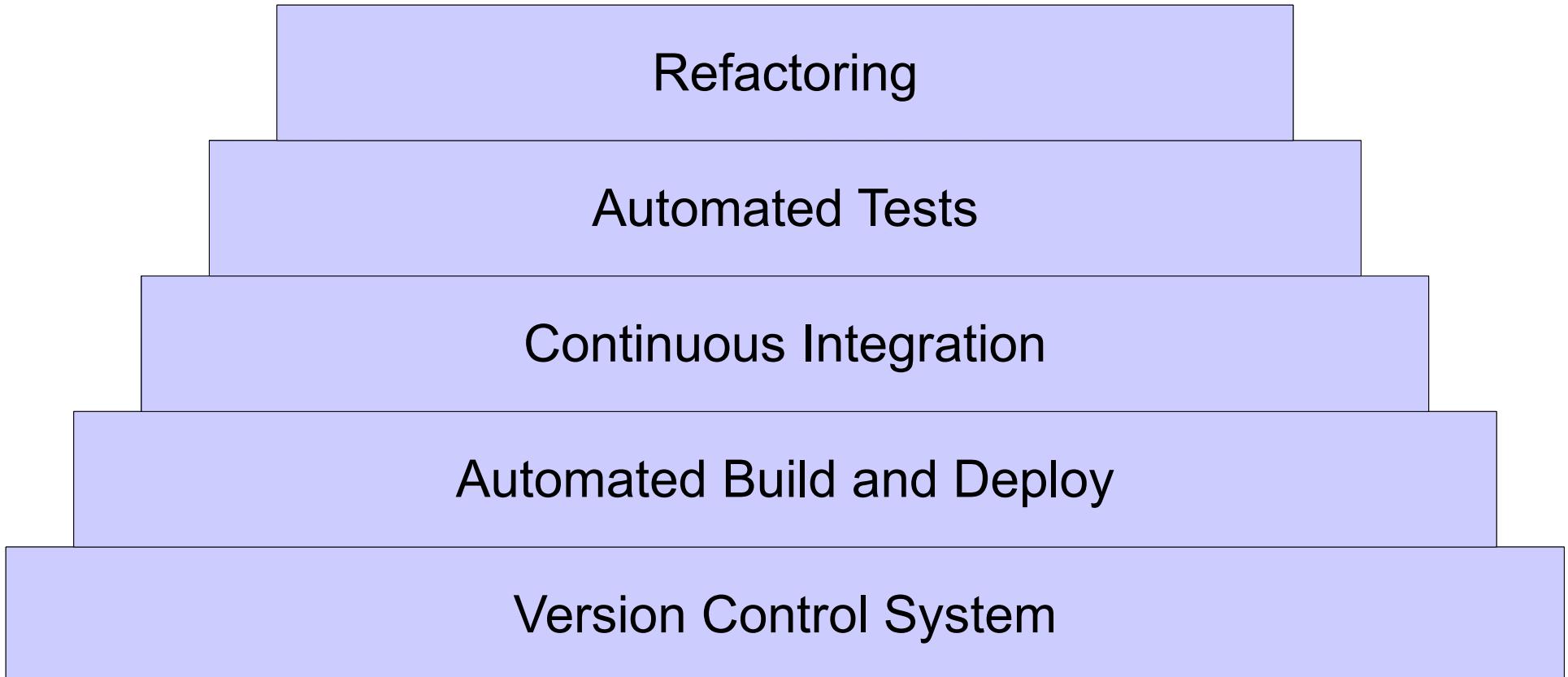


TextTest

- Insert log statements at key points in the code
- Test by log comparison
- <http://texttest.org>

```
output.demo vs. output.demo - TkDiff 4.0
File Edit View Mark Merge Help
1 : 8c8 C:\TEXTTEST\TESTS\demo\AddDup\output.demo C:\UME\ALLTIP\LOCALS\Temp\demotmp03May145954\AddDup\output.demo
1 'set new movie name to' event created with arguments 'Star Wars'
2
3 'add movie' event created with arguments "
4 Adding movie 'Star Wars'. There are now 1 movies.
5
6 'add movie' event created with arguments "
7 !ERROR: 'Star Wars' has already been added!
8 !Adding movie 'Star Wars'. There are now 2 movies.
9
10 'close' event created with arguments "
11 Exiting the video store!
```

The Productivity Practices Stack



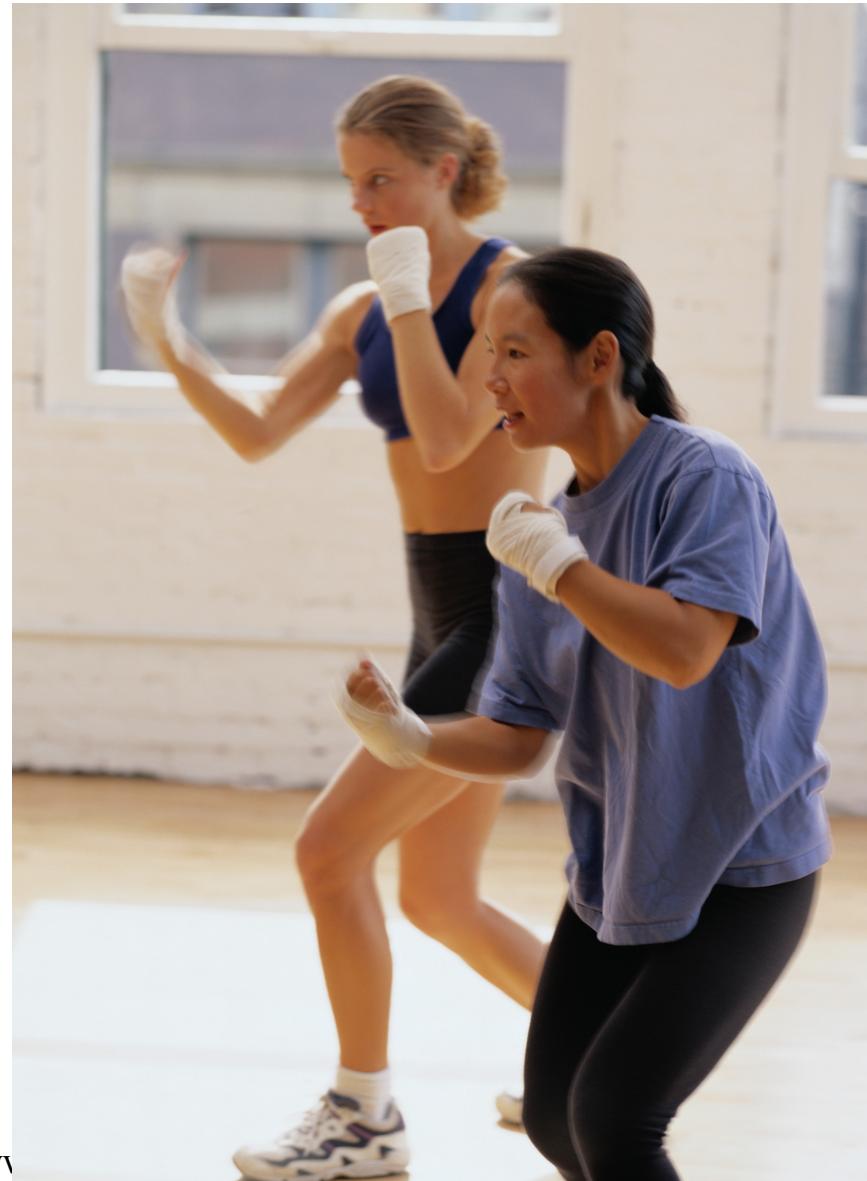
What are we refactoring *to*?

“There is no such thing as
good code and bad code,
just my code and your code”

What is clean code for **your team**?

The Coder's Dojo

A safe environment to
learn coding skills
such as
Test Driven Development
and writing
clean code

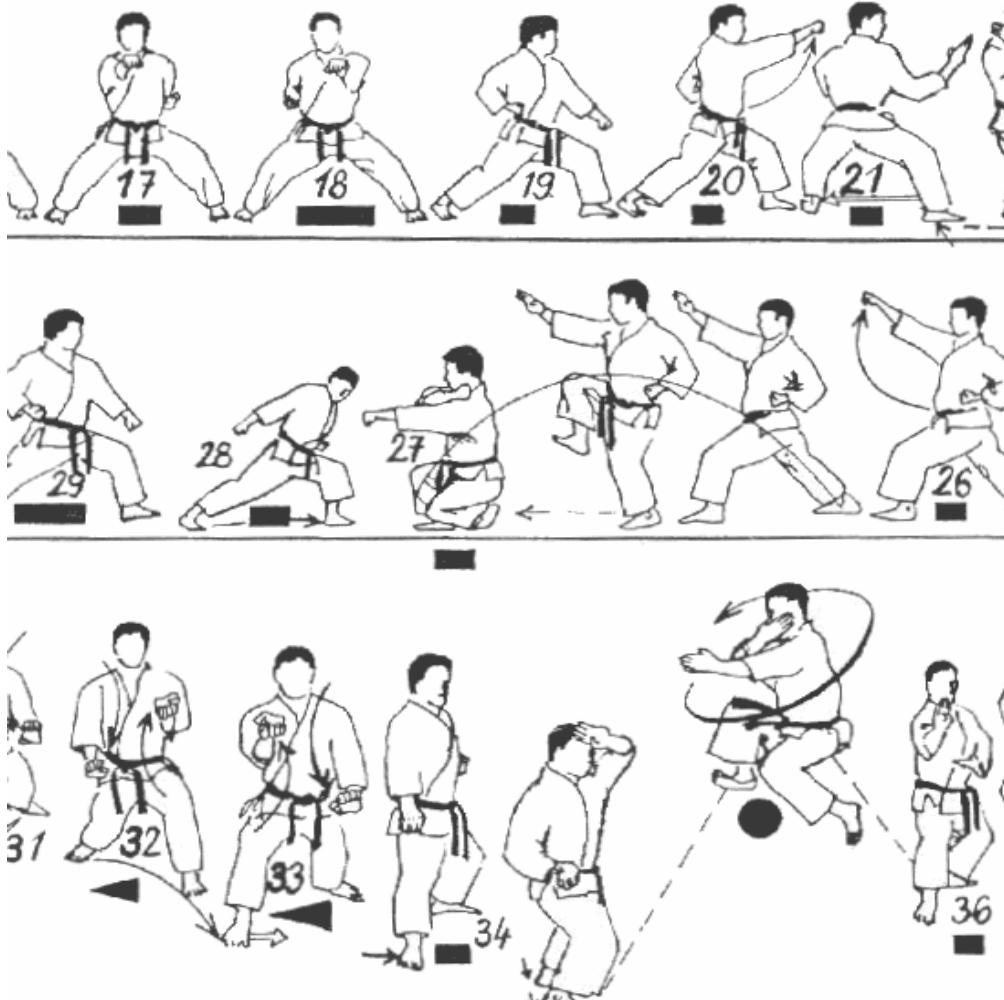


Kent Beck

*“I’m not a great programmer;
I’m just a good programmer with great habits”*



Code Kata



‘Pragmatic’ Dave Thomas put forward the idea of “code kata”

“A *kata* is an exercise in karate where you repeat a form many, many times, making little improvements in each.

the intent behind code kata is similar.”

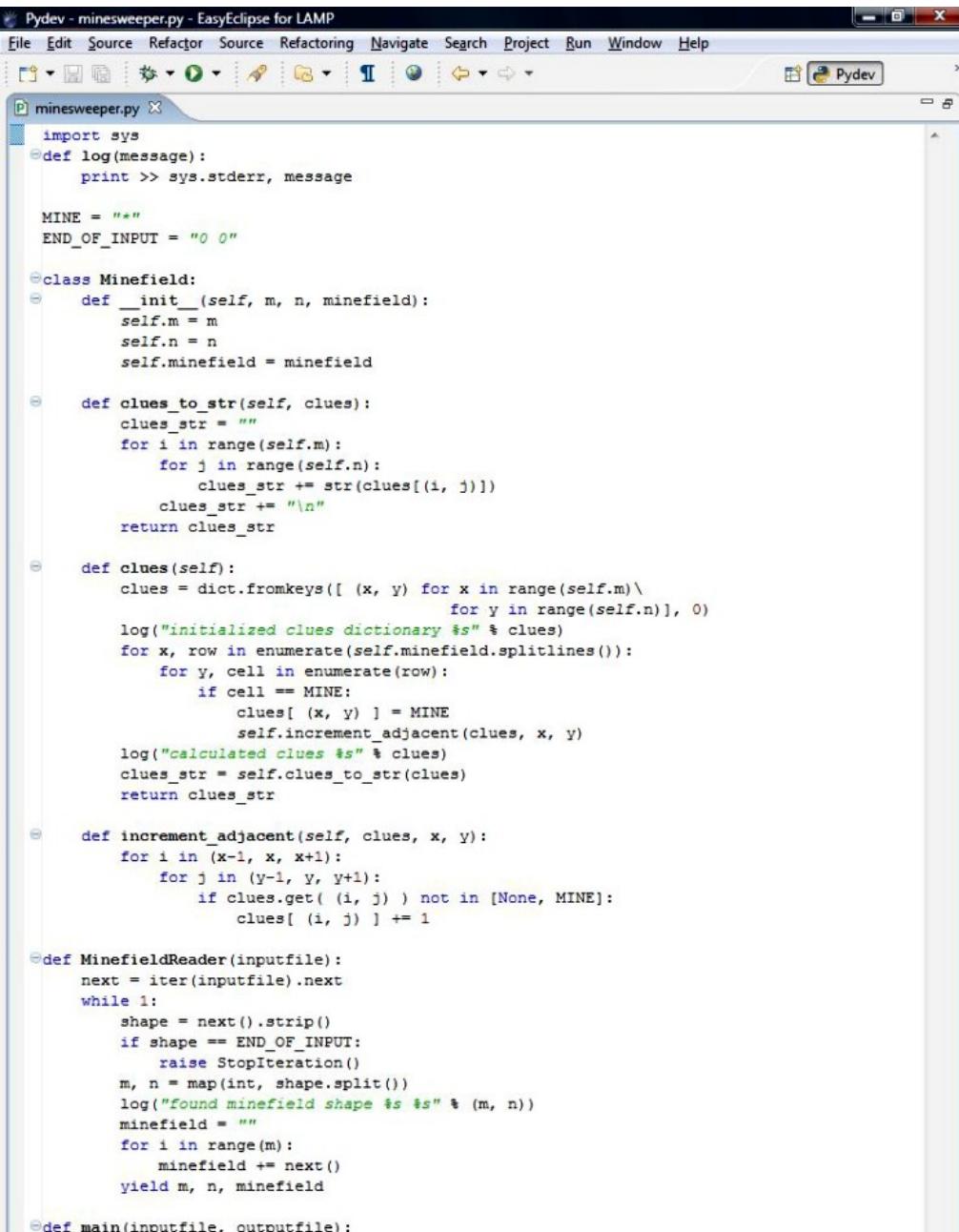
Code Kata in a group – The Coders Dojo

see <http://codingdojo.org>



- The dojo is the place where you go every week to practice and learn karate.
- At XP2005, Laurent Bossavit and Emmanuel Gaillot presented “The Coders Dojo”
- Their dojo continues to meet weekly in Paris.

What happens at a dojo meeting?



The screenshot shows the Pydev interface for Eclipse. The title bar says "Pydev - minesweeper.py - EasyEclipse for LAMP". The menu bar includes File, Edit, Source, Refactor, Source, Refactoring, Navigate, Search, Project, Run, Window, Help. The toolbar has icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, and others. The Pydev logo is in the top right. The code editor window contains the following Python code:

```
Pydev - minesweeper.py - EasyEclipse for LAMP
File Edit Source Refactor Source Refactoring Navigate Search Project Run Window Help
Pydev
minesweeper.py X
import sys
def log(message):
    print >> sys.stderr, message

MINE = "*"
END_OF_INPUT = "0 0"

class Minefield:
    def __init__(self, m, n, minefield):
        self.m = m
        self.n = n
        self.minefield = minefield

    def clues_to_str(clues):
        clues_str = ""
        for i in range(self.m):
            for j in range(self.n):
                clues_str += str(clues[(i, j)])
                clues_str += "\n"
        return clues_str

    def clues(self):
        clues = dict.fromkeys([(x, y) for x in range(self.m)\n                           for y in range(self.n)], 0)
        log("initialized clues dictionary %s" % clues)
        for x, row in enumerate(self.minefield.splitlines()):
            for y, cell in enumerate(row):
                if cell == MINE:
                    clues[(x, y)] = MINE
                    self.increment_adjacent(clues, x, y)
        log("calculated clues %s" % clues)
        clues_str = self.clues_to_str(clues)
        return clues_str

    def increment_adjacent(self, clues, x, y):
        for i in (x-1, x, x+1):
            for j in (y-1, y, y+1):
                if clues.get((i, j)) not in [None, MINE]:
                    clues[(i, j)] += 1

def MinefieldReader(inputfile):
    next = iter(inputfile).next
    while 1:
        shape = next().strip()
        if shape == END_OF_INPUT:
            raise StopIteration()
        m, n = map(int, shape.split())
        log("Found minefield shape %s %s" % (m, n))
        minefield = ""
        for i in range(m):
            minefield += next()
        yield m, n, minefield

def main(inputfile, outputfile):
```

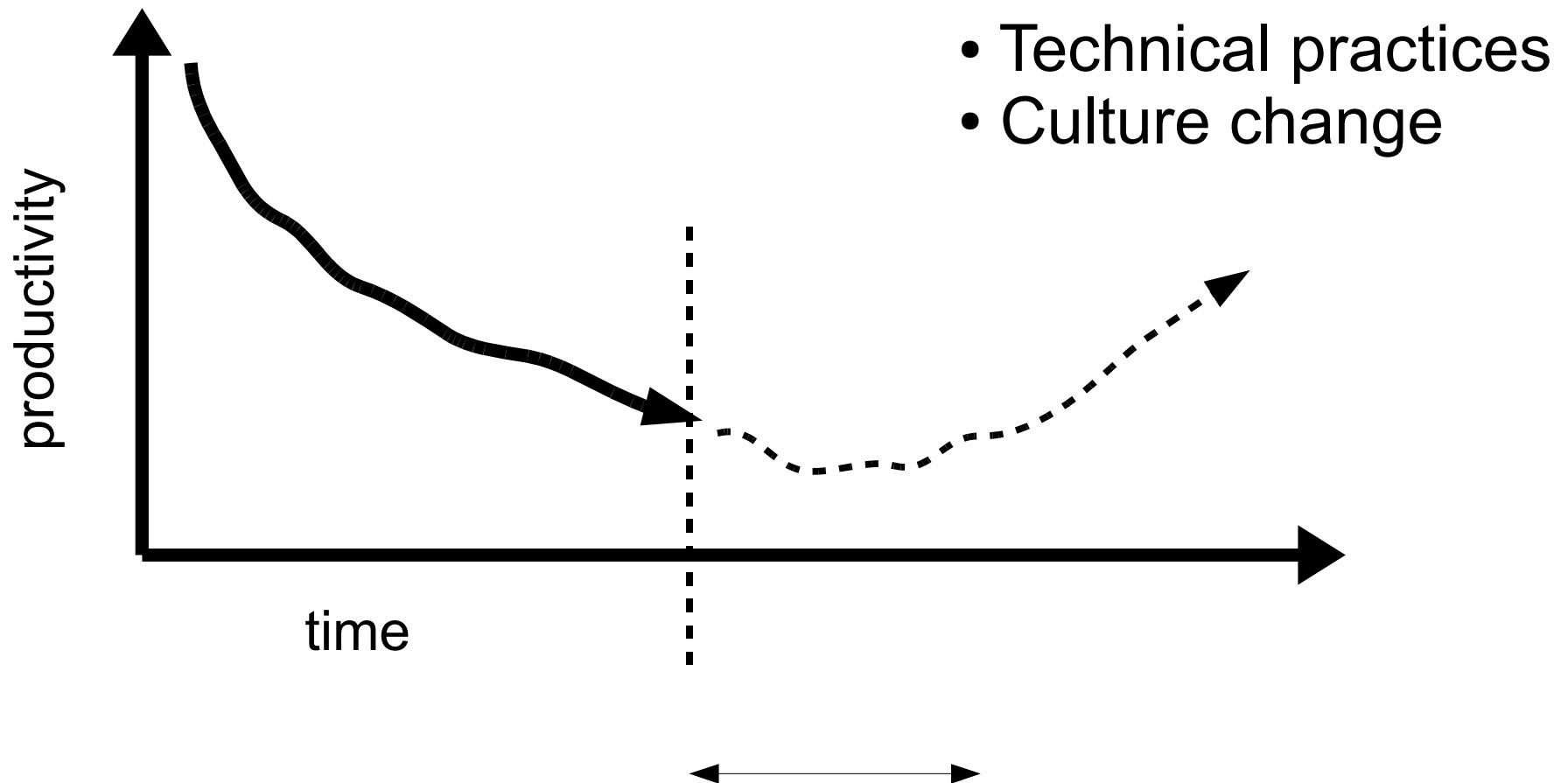
- 6 – 15+ programmers meet as equals.
- One person organizes/ moderates, and presents the code Kata to be solved.
- Solve the kata
- Learn from others, teach what you know
- At the end, hold a short retrospective.

Join or Start a Dojo

- There are coding dojos all over the world.
- <http://codingdojo.org>
- Why don't you go along to one and see if you can teach or learn something?

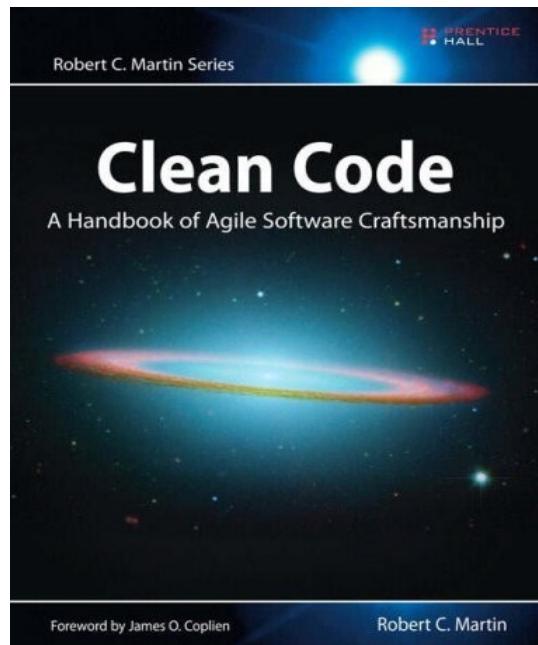


Productivity turnaround

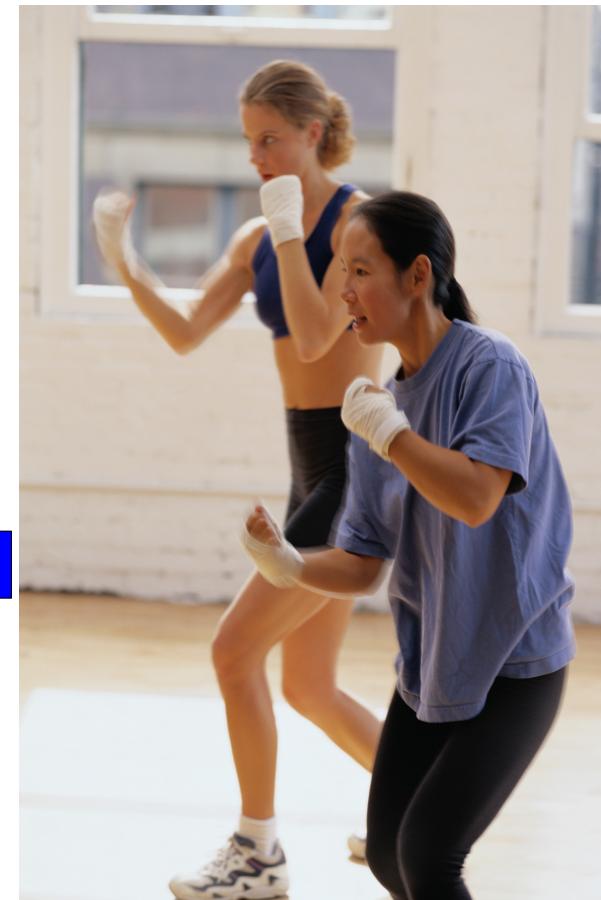


Productivity drops further
www.jsolutions.se

Take home messages



Refactoring
Automated Tests
Continuous Integration
Automated Build and Deploy
Version Control System



Questions?

- <http://www.codingdojo.org>
- <http://texttest.org>
- <http://checkstyle.sourceforge.net/>
- <http://emilybache.blogspot.com/>