TDD introduction

PRAGMATISTS

TDD promises

better design

less debugging

dynamic work

quality

safeness

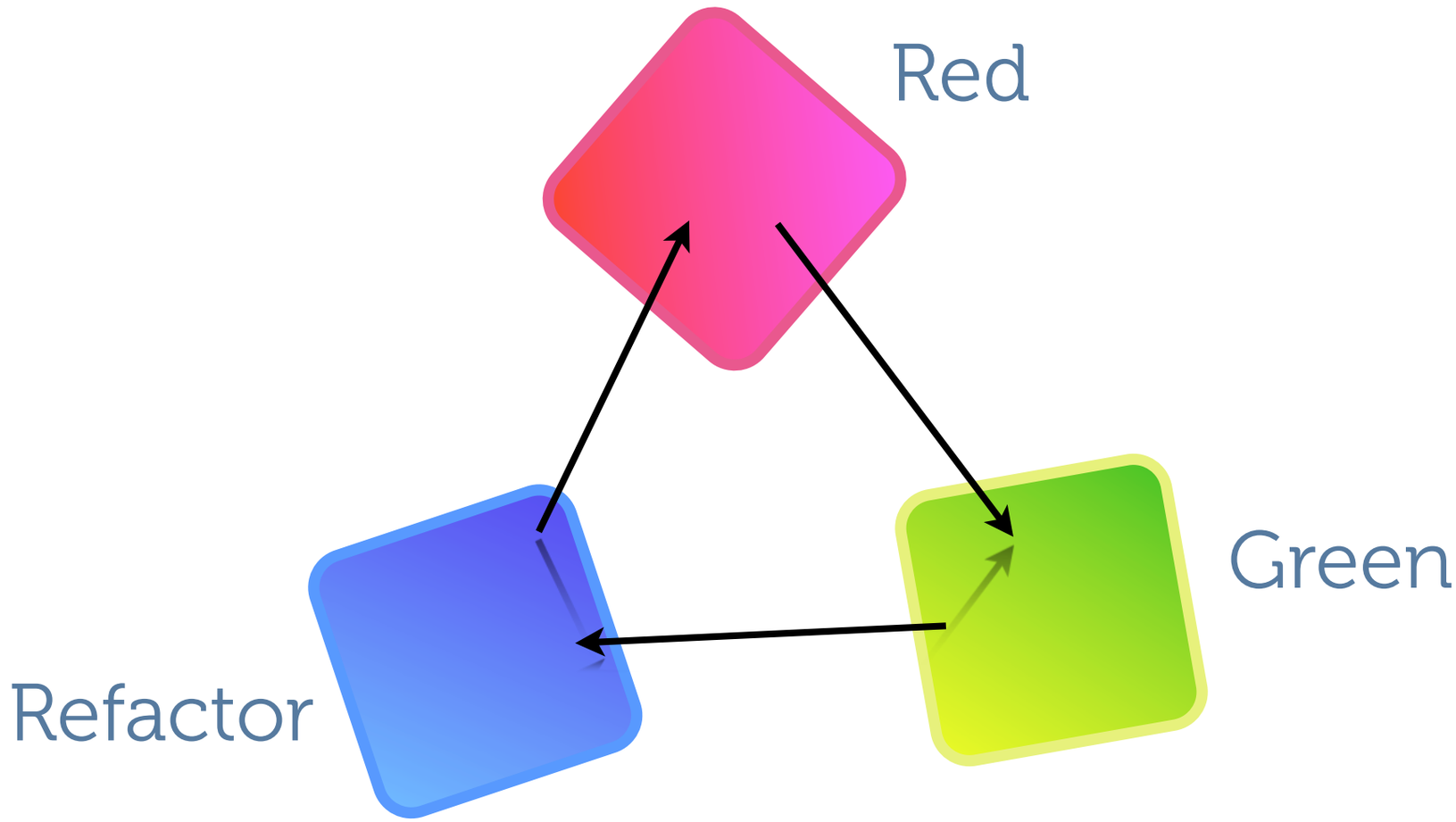
courage to refactor

durability

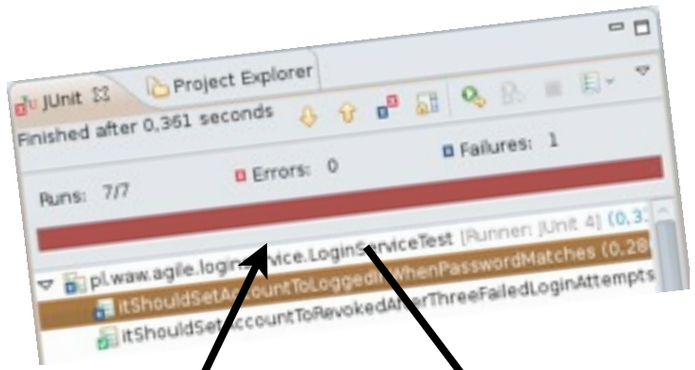
executable specification

effectiveness

A technique of software development based on repeating a short cycle:

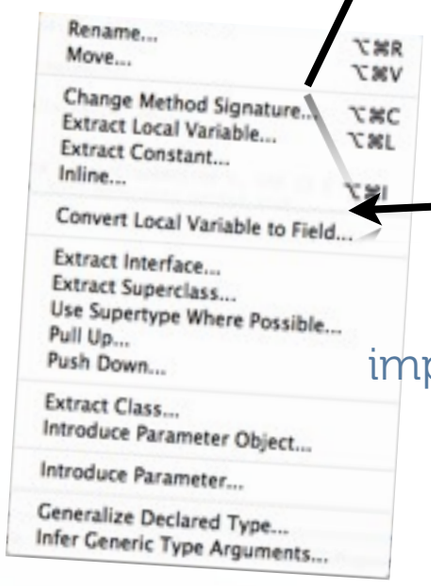


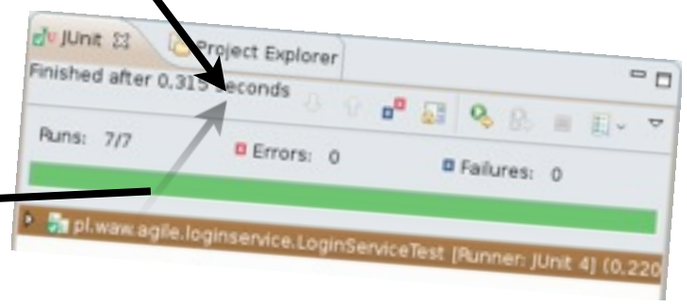
A technique of software development based on repeating a short cycle:



write new, unpassing test

make the test pass





improve the design and code readability

Red

What should be the behaviour?

```
Clock clock = new Clock();
clock.set(6, 15);

wait(5).minutes();

assertClockAt(6, 20);
```

Red

- thinking - which way to choose?
- designing

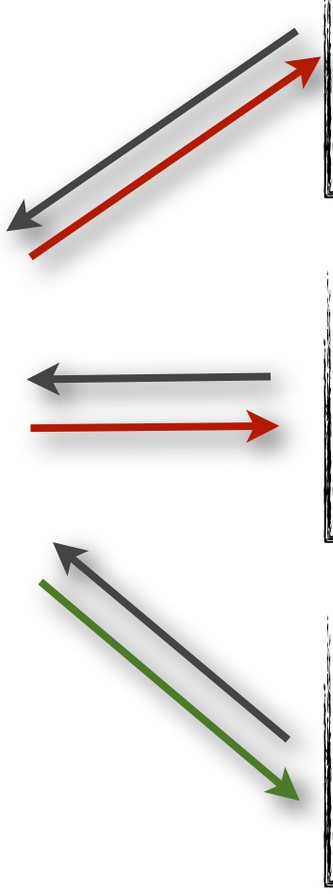
```
Clock clock = new Clock();
clock.set(6, 15);

wait(5).minutes();

assertClockAt(6, 20);
```

- code quality
- testability
- test quality

Red ➔ Green

How to achieve the behaviour?

```
Clock clock = new Clock();
clock.set(6, 15);

wait(5).minutes();

assertClockAt(6, 20);
```



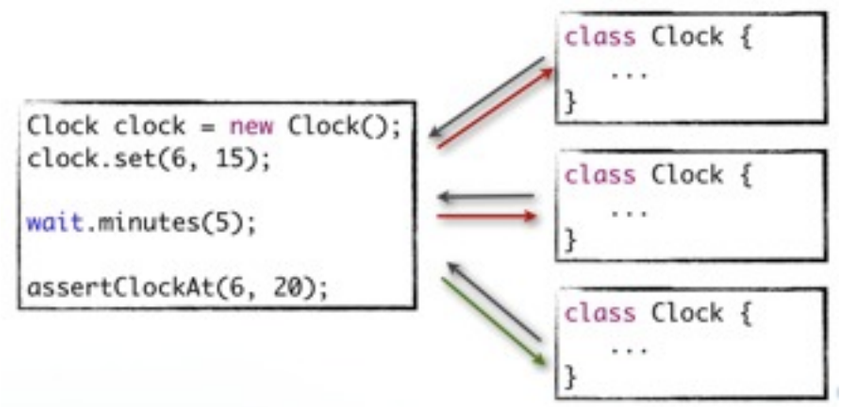
```
class Clock {
    ...
}
```

```
class Clock {
    ...
}
```

```
class Clock {
    ...
}
```

Red ➔ Green

- programming

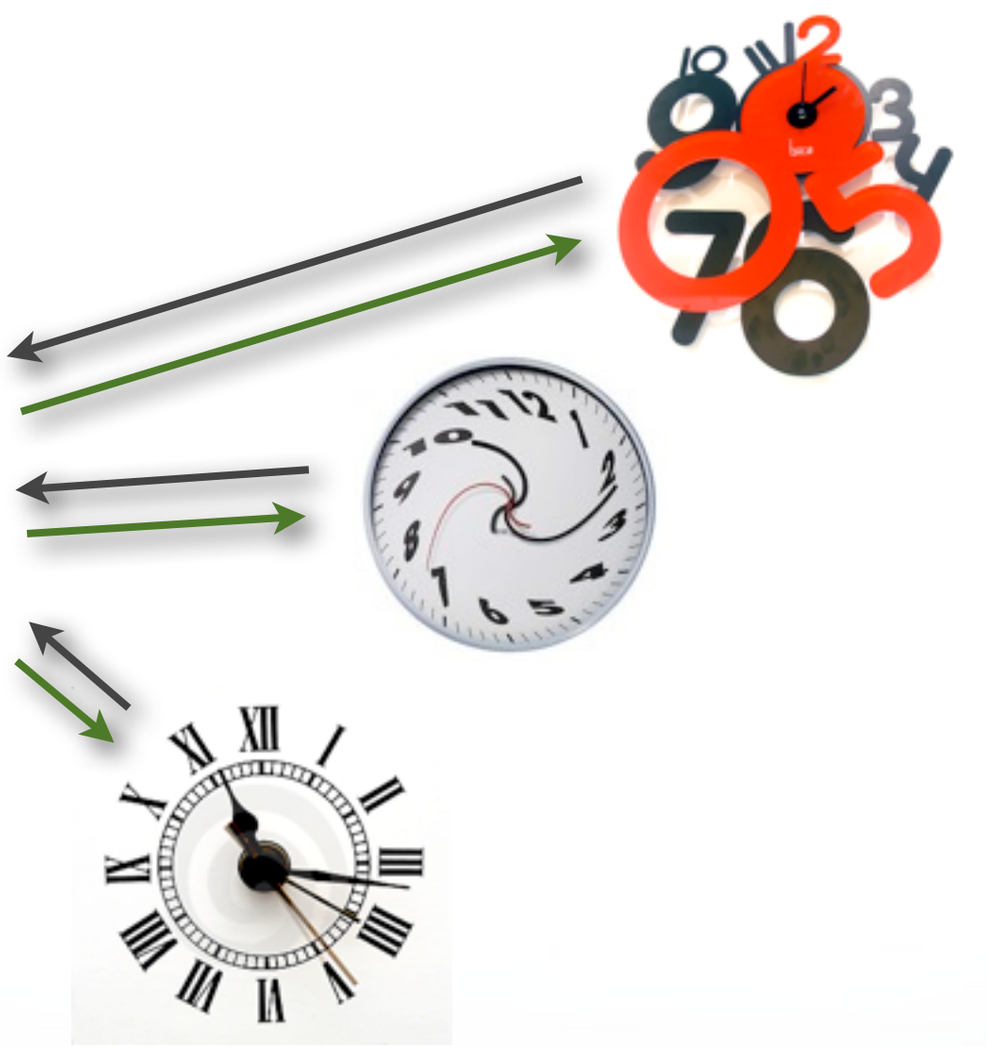


- is the new code working?
- didn't I break any other piece of code?
- did I choose the right test size?
(coding time)

Refactor

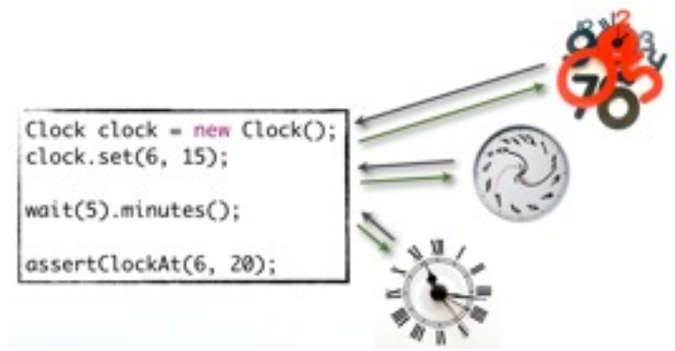
How should the code look like?

```
Clock clock = new Clock();
clock.set(6, 15);

wait(5).minutes();

assertClockAt(6, 20);
```



Refactor

- designing
- refactoring



- design quality (how easy it is to refactor)
- tests' quality (how safe is it to refactor)

demo

Why using TDD?

- WORKING code
- fast feedback
- easier to find additional corner cases
- better design
- less debugging

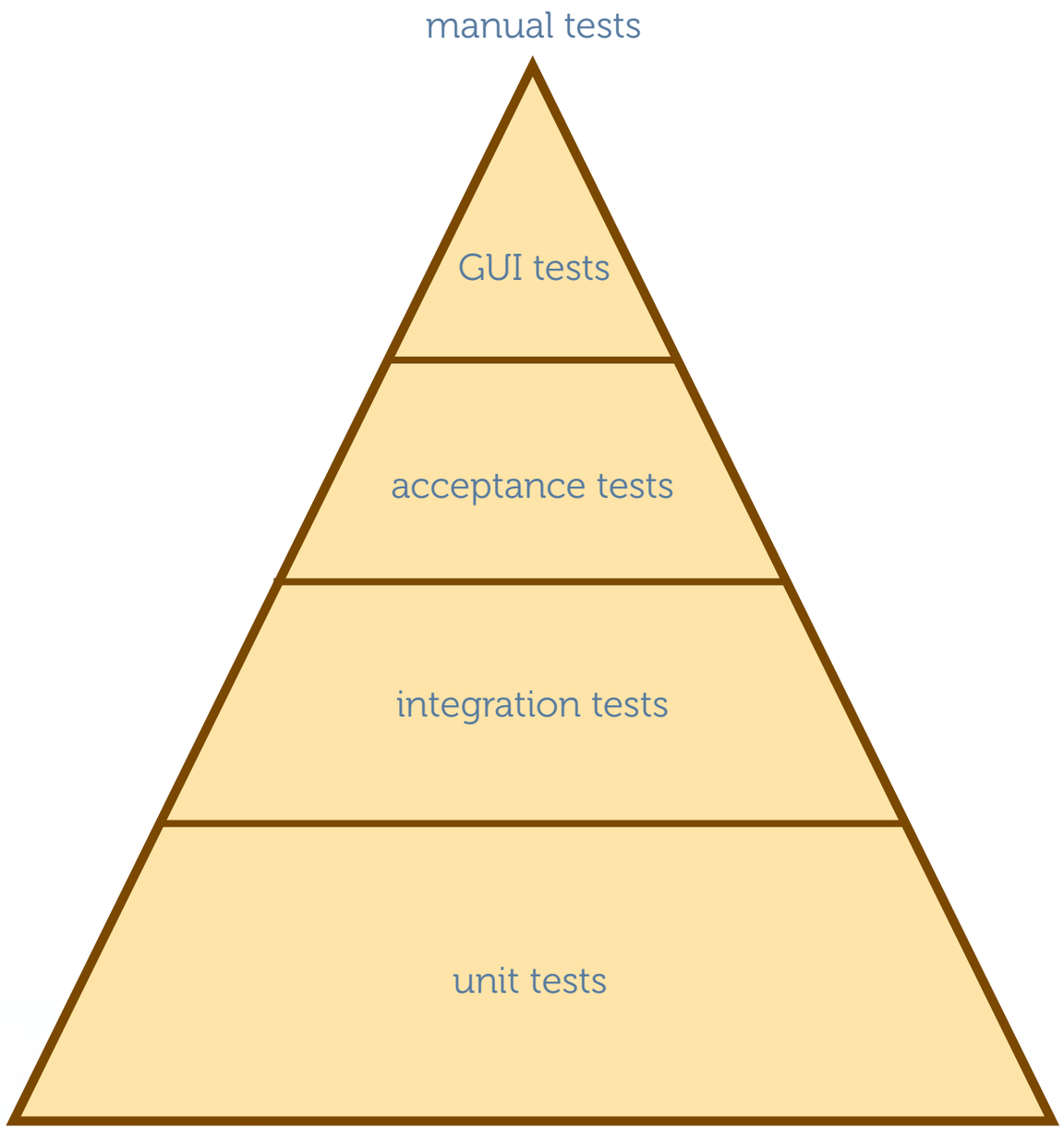
„side effects"

- testability
- ... which gives lower coupling
- better understanding of the domain and acceptance criteria
- executable specification / always up-to-date code documentation

testing levels

- unit
  Do objects do what they should, and does it feel comfortable to work with them?
- integration
  Do layers, modules, components correctly cooperate?
- acceptance
  Is a requirement correctly implemented? Does the whole system work?

pyramid of tests



history

- started around 1999
- Kent Beck
- one of the obligatory XP practices