1

How did it feel to work with such fast, comprehensive tests?

Some time was spent to understand the code

Difficult at the beginning, but felt safe to do changes

Easy

Understandable

Safer than without tests

They were a safety net while changing code

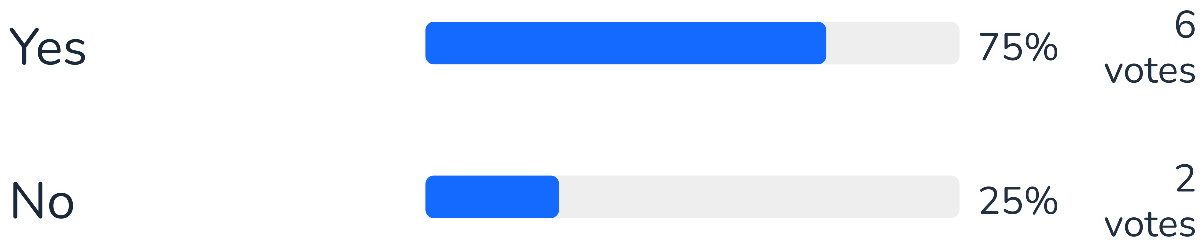
less frustrating

Anxious

Very efficient

2

Did you make mistakes while refactoring that were caught by the tests?



3

Did you ever make any refactoring mistakes and then back out your changes? How did it feel to throw away code?

Didn't happen to me, but I guess it would be totally fine to throw away some code if it's not working

waste of time

Frustrating

4

What would you say to your colleague if they had written this code?

At least read the Intellij suggestion

The code was good but it needed some refactoring

Use more meaningful names, I spent too much time trying to understand the code

Probably that it could be refactored in a better way

follow the software development course

Pick more intuitive names for variables and methods

5

Have a look at the refactored version of TennisGame1

https://github.com/dario-campagna/Tennis-Refactoring-Kata/blob/refactoring/java/src/main/java/TennisGame1.java

Does the refactored code communicate its intention better than the original code? Why?

There are no nested statement

Yes, much easier to read

Meaningful names, single task methods

Easy to read

Yes it is more understandable

It's better because methods have meaningful and intuitive names

♥ 1

Yes, has more intuitive names

Really straightforward to understand

What are the main differences between the refactored code and the original one?

Some methods simply apply another method

Better names and shorter methods

Methods and variable names that tell the usage

The refactored one has more compact methods

Smaller function in the second one
Less if else statement
Better names

The refactored code has many more methods, with better names

Better names are used for variables and methods

What would you change in the refactored code?

Maybe add a "Player" class where to save name and score could be useful

Maybe a little bit to much function, for example two of them are used only once in the code

Try to make the class shorter

Maybe there are too much methods and you constantly have to jump from one method to the other

Methods names are meaningful, but maybe some of them are too long