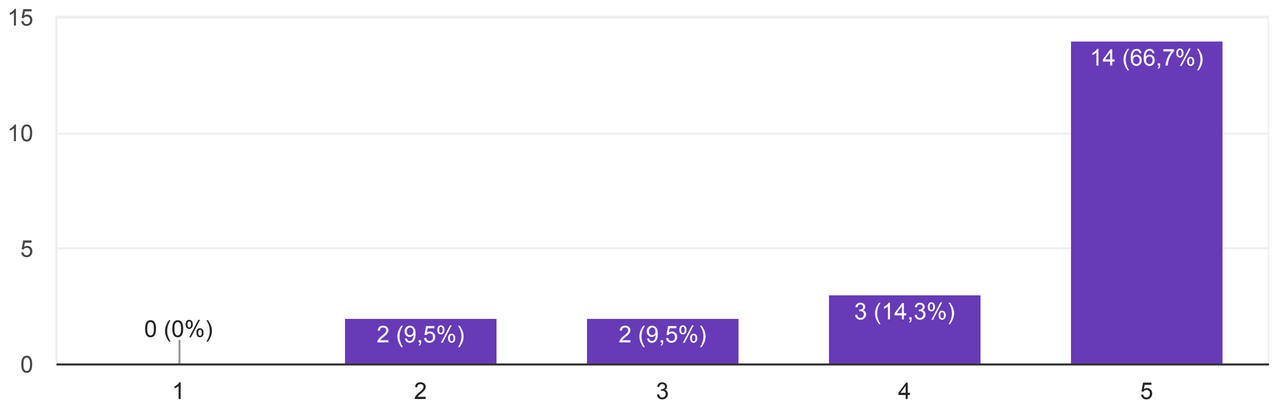
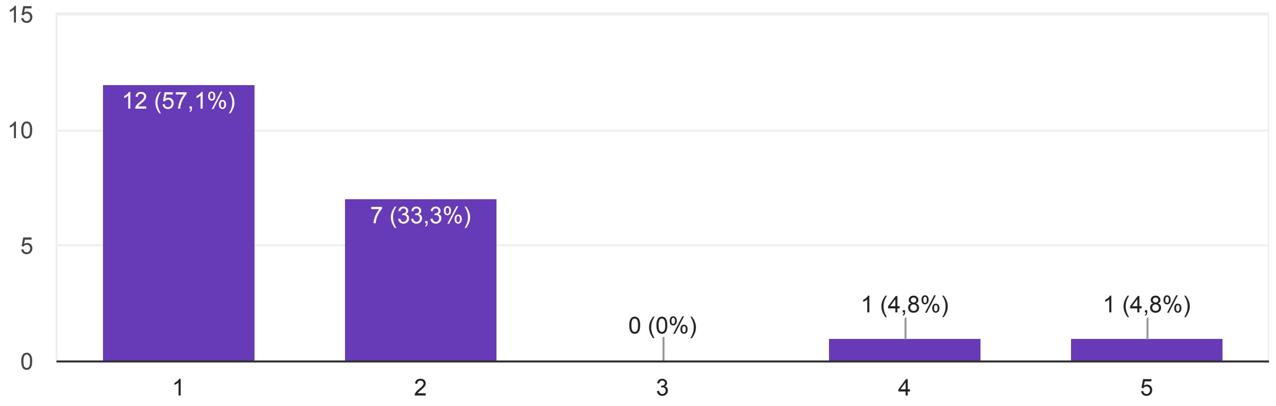


How would you rate the difficulty to understand the provided test smell descriptions at the method level?

21 risposte



Would you like to modify the Smell Description on the method level? If yes, please specify in brief.

4 risposte

I think they are fine and useful for better understanding test quality. I would leverage the extracted static information and descriptions for guiding the fixing with potential patches.

I think the descriptions are precise and clear, easy to follow.
However, maybe the suggested fix would be helpful at method Level too. I understand, that they are on class Level to allow

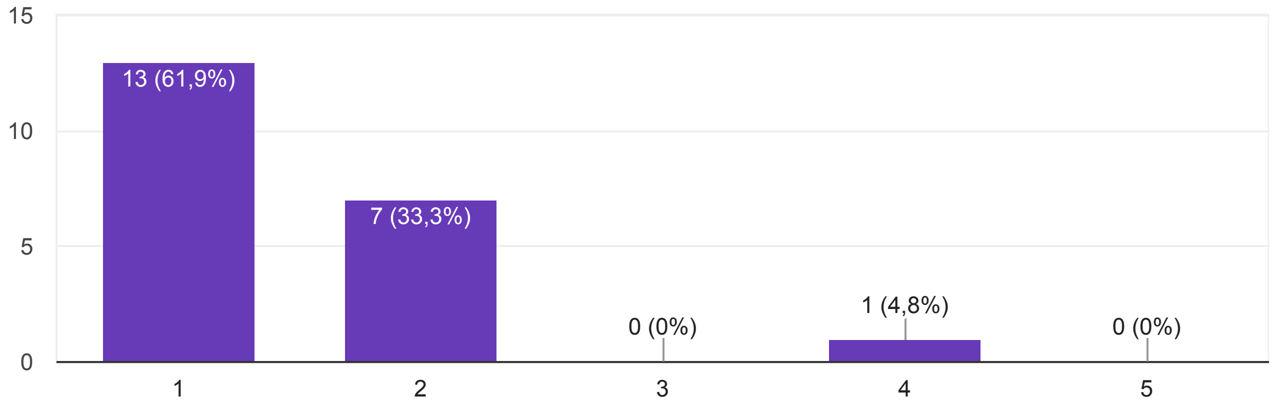
bundling but it was a bit of a nuisance having to scroll back to the top to see the Suggestion.

I think a link to some patches for fixing the problem would be also useful to have...

No

How would you rate the difficulty to understand the provided test smell descriptions on the class level?

21 risposte



Would you like to modify the general descriptions (on the class level) of the smell? If yes, please specify in brief.

5 risposte

No.

no.

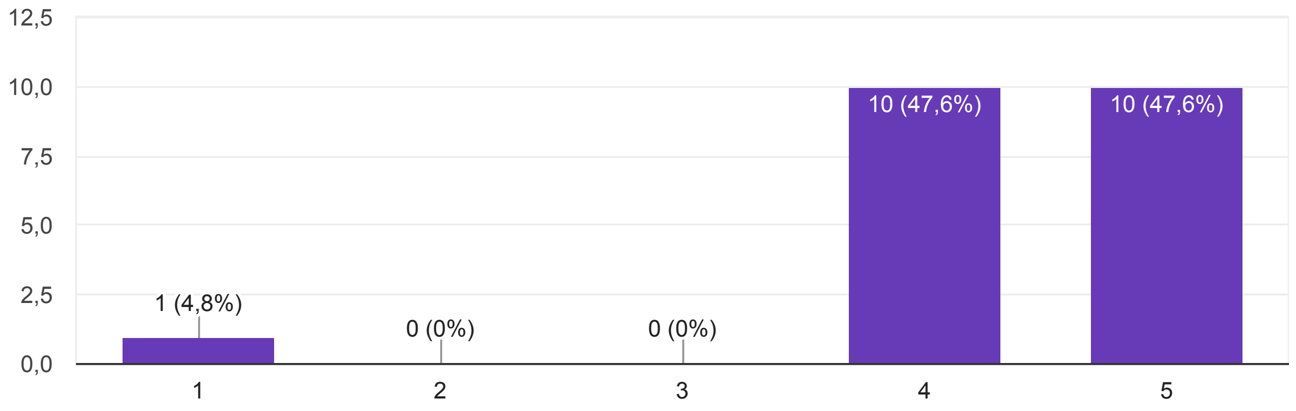
probably better to provide some more information about the class under test too

I guess the most important think to modify is the part related to the description on the smells in the project. How critical are they for developers (e.g., does it they impact they productivity)?

give more context information

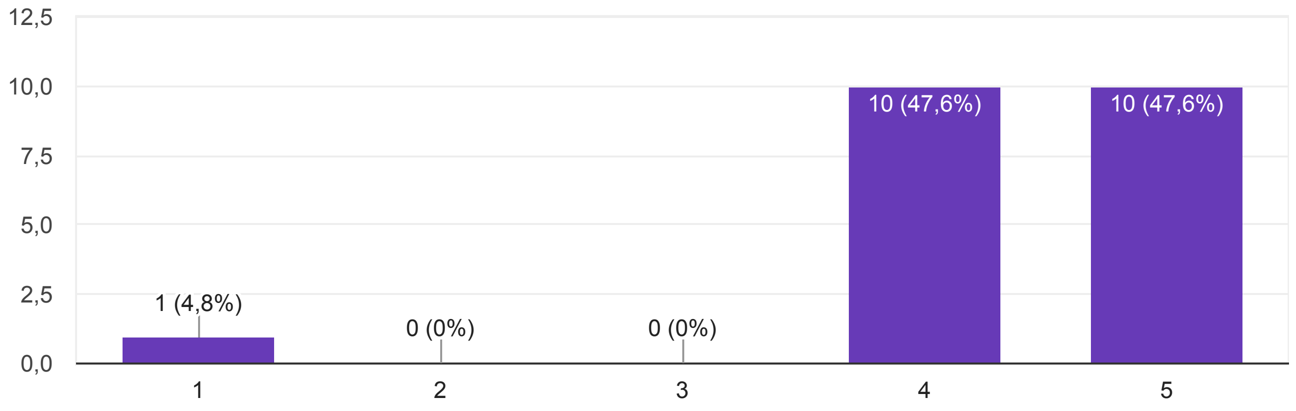
Are the generated Test Smell Summaries useful to be more aware of the general test quality?

21 risposte



Modifying the test code as suggested by the description provided by TSD is helpful?

21 risposte



What type of TSD descriptions do you find more relevant/useful for the performed

tasks?

12 risposte

I doubt the described smells are actual smells without understanding the semantics of the test. Just splitting methods in several pieces is not a real solution. It actually adds more lines of code to the test, more methods, more signatures - could actually make the code more complicated to maintain. The same goes for creating data classes, which now need to be maintained as well, and especially in Java, where constructors need to be written, this adds a lot of bloat.

I substantially found useful to have the comment in the actual place where smells are located + i FOUND HELPFUL THE PROVIDED RECOMMENDATIONS

The test method contains too many things.

I think the class Level TSD provide a good overview of the classe's Problems.

The descriptions at both levels serve important purposes.

the combination of class and method descriptions are useful

the one at the method level (I would mote the refactoring description at the method level, leaving the more general one at test class level).

all of them, but i think some changes can be approved to make them more actionable, especially with what concern the integration of potential patches

first the one at test method level, only after the one at class level

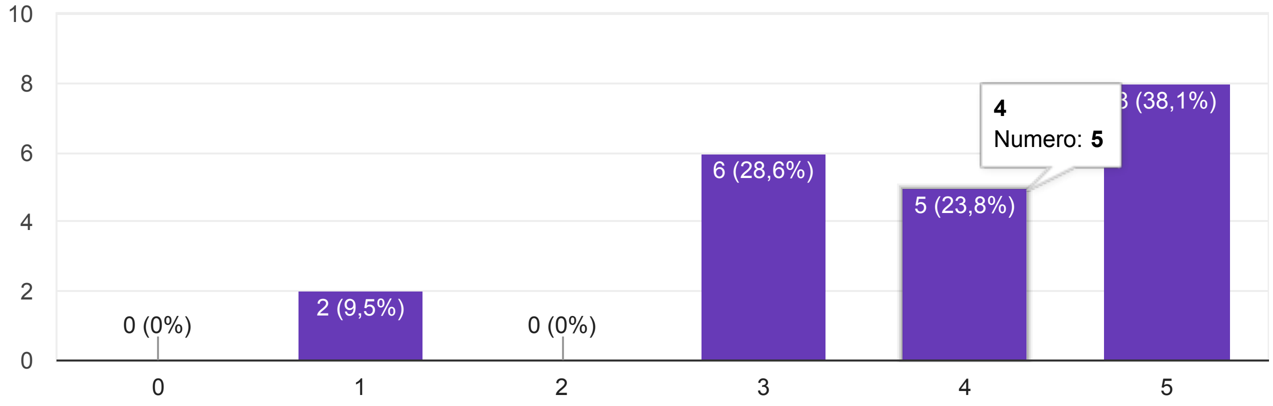
the one at the method level

the actual combination of class and method descriptions

class level description. It gives me the emphasis of attention quickly.

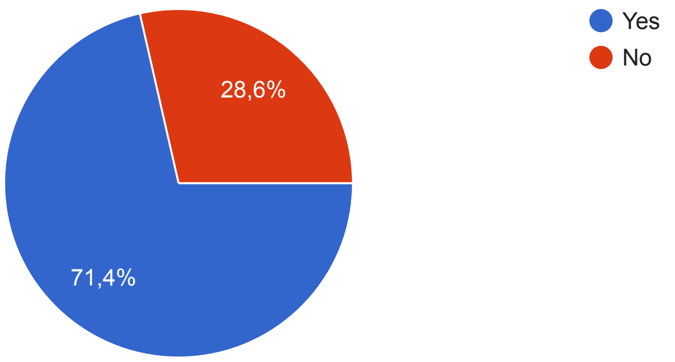
Impact of test case summaries on your perception of test cases.

21 risposte



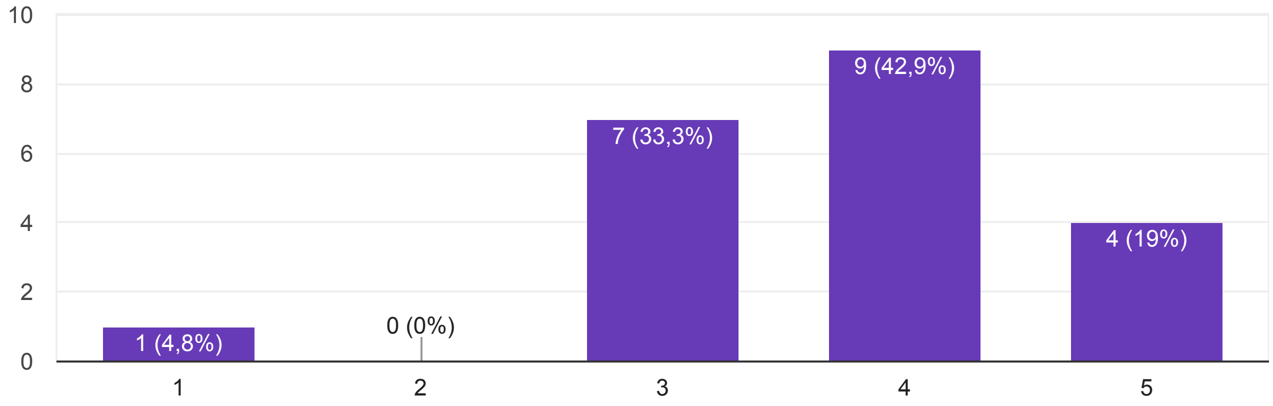
Was the given time enough to complete both tasks?

21 risposte



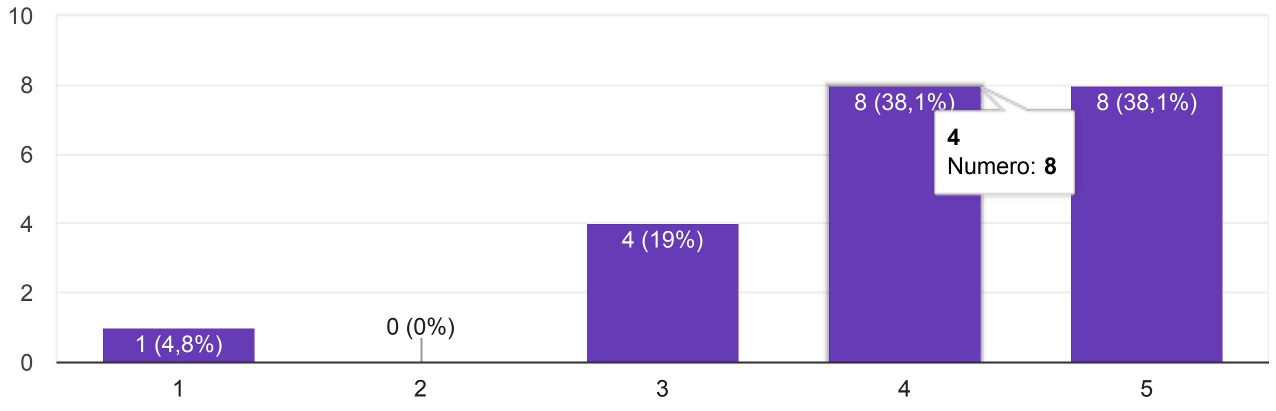
How difficult are both tasks to understand and complete?

21 risposte



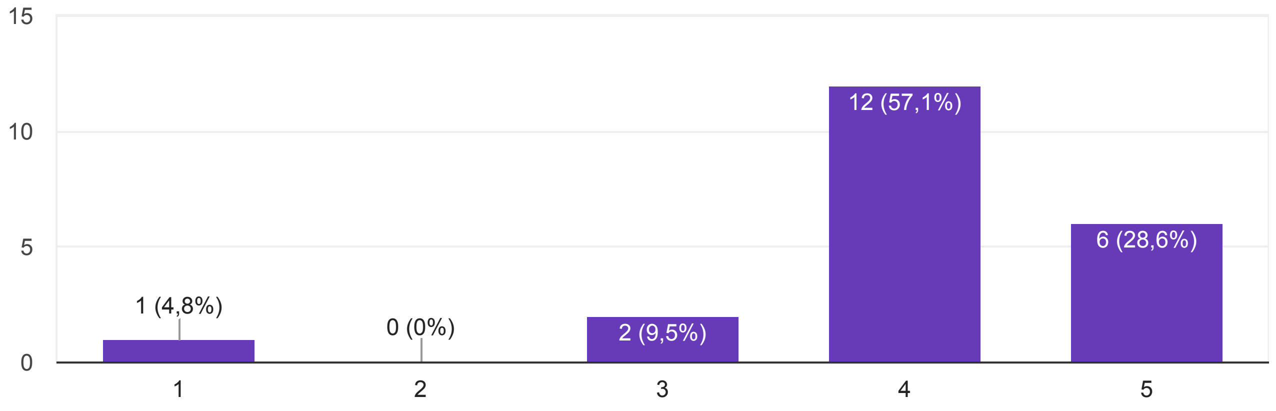
Modifying the classes leads to better and improved code?

21 risposte



The task without the generated comments/description is prohibitively difficult?

21 risposte



Considering only the content of the generated comments/description and not the way it was presented, the description

21 risposte

● is not missing on any information.
● is missing on some information.
● is missing on some very important information.

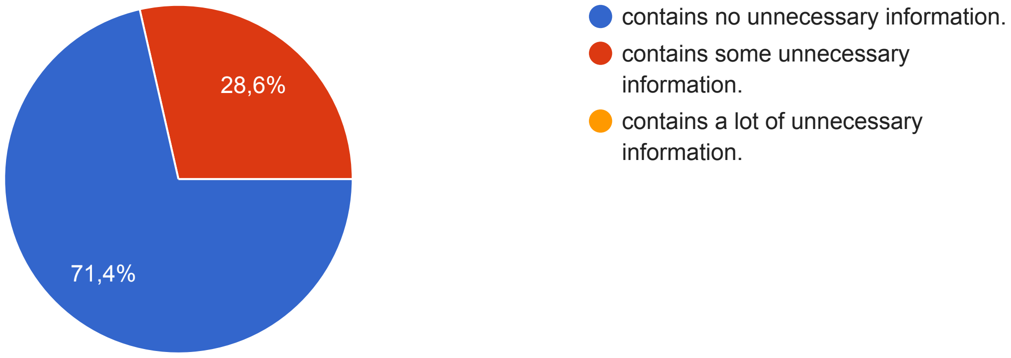
If you feel we have missed out on some important information in the descriptions, please specify the information below.

1 risposta

Suggestion on how I can split the code to reduce the size of the method. For instance, by grouping based on the used variables. Suggestion on which parameters to remove. For instance, if some (parameter) values are redundant and may be deduced from other parameter values.

Considering only the content of the generated comments/description and not the way it was presented, the description

21 risposte



If the generated comments/description has unnecessary information, please specify.

5 risposte

The descriptions are a little bit redundant in general, but still okay for someone who knows nothing about code smells - but then again, if someone knows nothing about code smells, maybe they shouldn't be the ones changing the code. It would just be a mechanical application of the suggestions without any real meaning behind them.

Description of the method arguments is unnecessary.

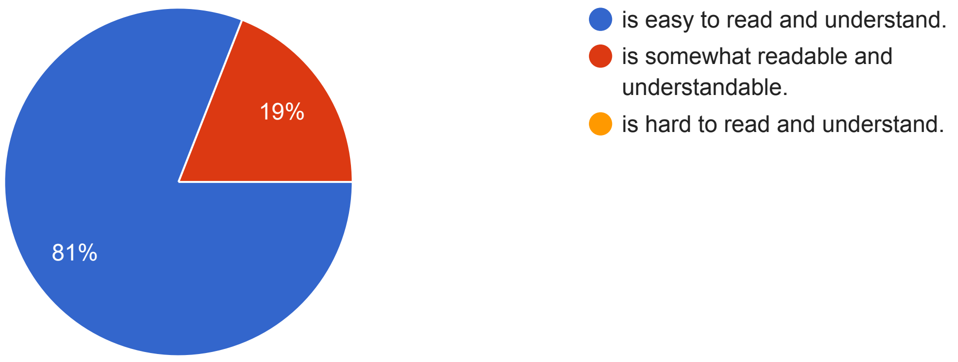
The percentages shown in the description are not useful to refactor test smell.

Not sure whether knowing the percentage provides much benefits.

"A long list of parameters is hard to understand, to use and may become
inconsistent over time." we donot need this to tell me the danger of code smells

Considering only the content of the generated comments/description and not the way it was presented, the description

21 risposte



Do you have any suggestions to improve the readability of generated comments/description? If yes, please specify.

4 risposte

-

the one at the method level (I would mote the refactoring description at the method level, leaving the more general one at test class level).

The most important think to modify is the part related to the description on the smells in the project. How critical are they for developers (e.g., does it they impact they productivity)?

Make a simplified ranking of code smells overally. Give details when user wants to know how to fix it.

If you have any other comment to make, please make use of the below space.

2 risposte

The study itself suffers from an important drawback: one of the two java files contained comments, the other did not - yet it was very, VERY easy to guess which suggestions TSD would have suggested where the comments were missing, i.e. "more than 6 parameters" and "method too long". After reading/correcting the first java file containing comments, it was obvious what was expected in the files missing the suggestions - if everyone guesses this, you won't have any differences whether TSD suggestions are present or not...

Maybe it would be good to indicate that the solutions should be submitted at the beginning of the experiment. Unfortunately, I am filling the survey with few days delay and at the moment I do not have my modifications on the code any more.

If you have any concerns and issues regarding the tasks, please don't hesitate to contact us.

Thank you very much for all your effort and time.