

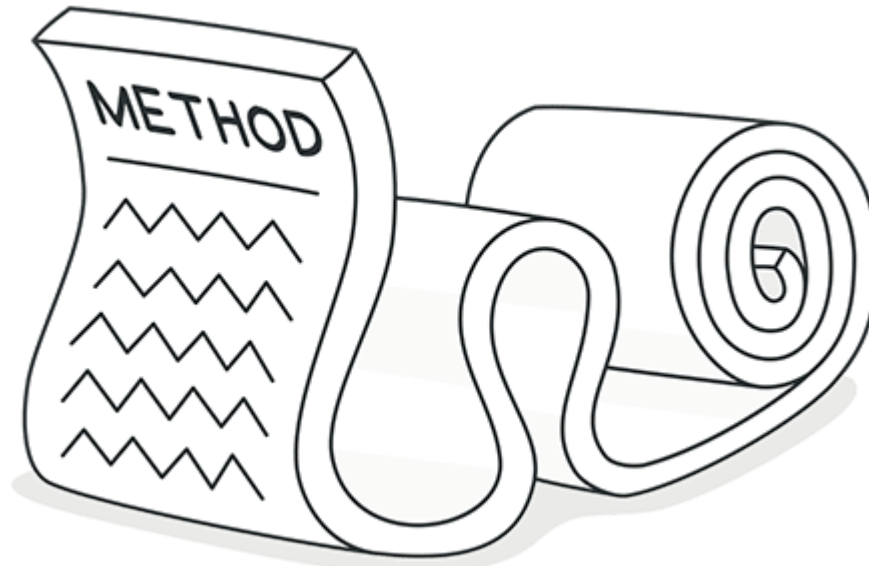


[Home](#) / [Refactoring](#) / [Code Smells](#) / [Bloaters](#)

Long Method

Signs and Symptoms

A method contains too many lines of code. Generally, any method longer than ten lines should make you start asking questions.



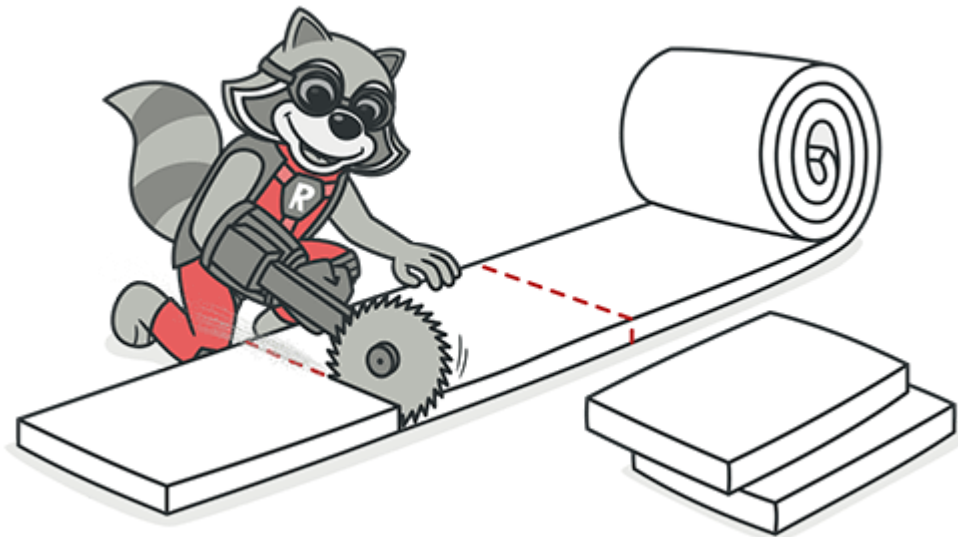
Reasons for the Problem

Like the Hotel California, something is always being added to a method but nothing is ever taken out. Since it's easier to write code than to read it, this “smell” remains unnoticed until the method turns into an ugly, oversized beast.

Mentally, it's often harder to create a new method than to add to an existing one: “But it's just two lines, there's no use in creating a whole method just for that...” Which means that another line is added and then yet another, giving birth to a tangle of spaghetti code.

Treatment

As a rule of thumb, if you feel the need to comment on something inside a method, you should take this code and put it in a new method. Even a single line can and should be split off into a separate method, if it requires explanations. And if the method has a descriptive name, nobody will need to look at the code to see what it does.

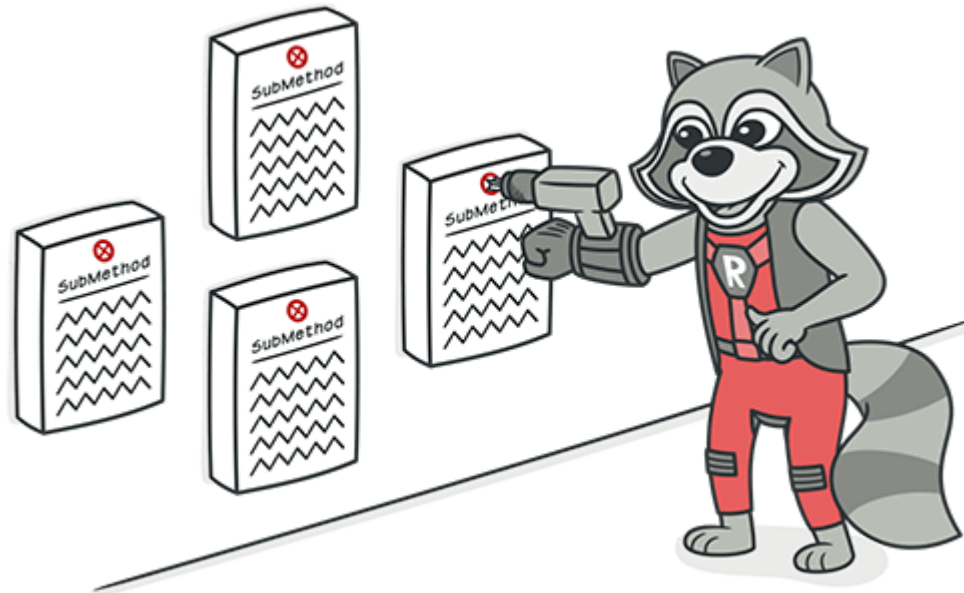


- To reduce the length of a method body, use **Extract Method**.

- If local variables and parameters interfere with extracting a method, use **Replace Temp with Query**, **Introduce Parameter Object** or **Preserve Whole Object**.
- If none of the previous recipes help, try moving the entire method to a separate object via **Replace Method with Method Object**.
- Conditional operators and loops are a good clue that code can be moved to a separate method. For conditionals, use **Decompose Conditional**. If loops are in the way, try **Extract Method**.

Payoff

- Among all types of object-oriented code, classes with short methods live longest. The longer a method or function is, the harder it becomes to understand and maintain it.
- In addition, long methods offer the perfect hiding place for unwanted duplicate code.



Performance

Does an increase in the number of methods hurt performance, as many people claim? In almost all cases the impact is so negligible that it's not even worth worrying about.

Plus, now that you have clear and understandable code, you're more likely to find truly effective methods for restructuring code and getting real performance gains if the need ever arises.



Tired of reading?

No wonder, it takes 7 hours to read all of the text we have here.

Try our interactive course on refactoring. It offers a less tedious approach to learning new stuff.

★ Let's see...