Useful code refactoring

Tom Rochette <tom.rochette@coreteks.org>

November 2, 2024 — 36c8eb68

1 Question

How can you tell between noisy and useful code refactoring?

2 Answer

The classical adage in software development is that “if it ain’t broke, don’t fix it”. Code may not be in an ideal state, but you should not focus on refactoring it if you aren’t working to change it to do something else or so that it can be used in other parts of the code. It is fine to do code refactoring from time to time, especially if you have free cycles and know about a few pieces of code that have been bothersome. However, spending time refactoring systems that are already working but a bit clunky may not be the best use of time, especially if you or your team don’t have free cycles to spare to review your changes. A better use of time may for example to help others on their tasks or to prepare future work so it goes smoothly.

In a business environment, a refactor also means implicating other developers to review your changes. As such, this introduces distractions in those developers that could be more focused.

In this case, whether a refactor or other changes to the code is noisy is a matter of timing.