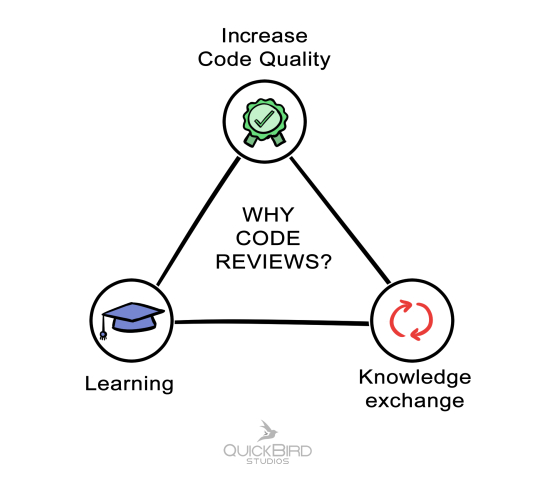
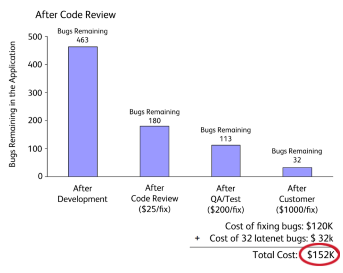
Code review

The main objective of the code review process is to evaluate all new code for bugs, errors and quality standards set by the organization. Every team can benefit from code reviews regardless of the development methodology. Agile teams, however, can realize huge benefits because the work is decentralized within the team. No one person is the only one who knows a specific part of the code base. Simply put, code reviews facilitate knowledge sharing within the code base and within the team. But the code review process should not be limited to one-way feedback. Therefore, an intangible benefit of the code review process is the collective improvement of the team's skills by exposing each other to a different approach to the topic at hand and thus being able to mentor new engineers by sharing our knowledge. This in turn reduces development costs through knowledge sharing and improves code performance because due to lack of experience, some young developers may not know the optimization techniques that could be applied to their code. The code review process allows these developers to gain skills and improve the performance of their code. In addition, it gives these young developers a chance to hone their skills and become experts in their field. The code reviews help create a certain cohesion within the team and bring a better communication on the project.





To summarize, here are the positive points of a code review

- Reduce costs

- Expose everyone to different approaches

- Minimize errors and their impact

- Better code quality

- Conform to style guidelines

- Have better estimates

- Mentor new engineers and share knowledge

- Ensure project quality

- Have a consistent implementation

- Improve code performance

- Create cohesion in the team

Let's talk about the bad uses of code reviews and the things to avoid during them.

One of the most important things in a team is cohesion and trust, it's important to know how to question each other and accept the objections of other developers. When we criticize something, we tend to highlight only the bad points and never the good. While a good code review is about pointing out imperfections in the code, your feedback should not be limited to finding errors. When you do reviews, you can also use comments to praise someone's good work. Do this especially if you learned something from the code or noticed a solution that you would not have found yourself. On the other hand, it is important not to hesitate to participate in code reviews, as some people are afraid to say what they think for fear of offending. In the way of saying things it is better for everyone to make suggestions rather than give orders such as, it allows others to come and ask themselves the question "what can it bring" rather than just execute the order. For example, if someone says "You should use getter and setter. This code is wrong. "This will discredit the developer's code and it may hurt the person's feelings, so it's better to offer a suggestion: In this case, "I would recommend using getter and setter, because..." This will allow the developer to research the topic and realize their mistake for themselves.

Resources:

[**https://hackernoon.com/code-review-for-real-people-261e3d9124ba**](https://hackernoon.com/code-review-for-real-people-261e3d9124ba)

[**https://tsh.io/blog/code-review-best-practices/**](https://tsh.io/blog/code-review-best-practices/)

[**https://www.atlassian.com/agile/software-development/code-reviews**](https://www.atlassian.com/agile/software-development/code-reviews)

[**https://www.brightspot.com/products/developer-life-5-reasons-why-the-code-review-process-is-critical-for-developers**](https://www.brightspot.com/products/developer-life-5-reasons-why-the-code-review-process-is-critical-for-developers)

[**https://phauer.com/2018/code-review-guidelines/**](https://phauer.com/2018/code-review-guidelines/)