Coding Standards and Best Practices

Today's world moves fast. Customers are demanding more features. Stakeholders want it done faster. This can add up to a lot of pressure for the ones who are actually building the service or product. The ones who are actually sitting at the keyboard and typing away.

Instead of bad coding habits, set yourself up for success with these coding best practices once you implement them, they'll save you and your team time, resources, and headaches.

By doing the following steps or following them, you'll have the best coding practices:

1. ADHERE TO YOUR STYLE GUIDE

Whether it's your organization's own style guide or the programming language's style guide, one of the most important coding best practices is to always follow your style guide. Every programming language has one. This style guide gives you all the details you need on how to indent code, where to place braces and spaces, naming conventions, and much more.

Read the guide thoroughly and take the basics to heart. You can always look up the less common requirements, but remember to always apply the rules of the style guide. Never cut corners when it comes to this area.

When you adhere to the rules of the style guide, other programmers can more easily read your code and work on it, if needed. If someone else is following the style guide, then you can easily jump into that project. This is a coding best practice because it allows for consistency throughout your code. When you follow the rules of the style guide, you can more easily find any mistakes or bugs.

Many organizations will modify style guides to match their own parameters. This can be a helpful way to make your team even more efficient. If you plan to do this with your team or company, make sure that your new guide is well thought out and that everyone on the team is aware of the customizations. Following the style guide's rules is one of the most vital and easy ways to implement coding best practices.

One tool that can help you and your team to stay on the same page in terms of style is Sider. Sider is an automated code review tool that analyzes your team's pull requests on the fly to help ensure that your code base is consistent. It can check for violations of style, as well as code quality, and it helps your team stay up to speed on best practices, as well as the rule-sets for each project.

2. MAKE SURE YOUR CODE IS READABLE

Instead of focusing on optimization, a coding best practice is to focus on code that is readable. Readable code is understandable code. Code that can be understood can be integrated more easily, allowing for greater efficiency in the project as a whole. As one of the coding best practices, always focus on readable code.

3. COMMENT AND DOCUMENT

When it comes to coding best practices, one habit that could help you is to start each function or method you create with a comment. In this comment, you can outline exactly what the function or method does, what its parameters are, and what it returns. You'll also want to describe any possible errors or exceptions. You can also summarize the major steps if your code is particularly complex, as well as outline the role of each file and class and any contents in each class field.

Doing this from the beginning keeps you consistent with commenting. You may be tempted to come back and add this in later, but the likelihood of you doing that is very low. There's a tendency to keep moving during your coding sessions.

In addition to any comments, be sure to document what it is your code does, as a whole. This can be a short and sweet README file, but this guide should explain exactly the actions that the code performs, including any dependencies, and it should contain instructions on how to build, install, test, and use your code. Comments and documentation help you to fulfill another of the coding best practices.

4. AVOID HARD – CODING

Hard coding in these cases is not technical debt, but could be seen as irresponsible with major consequences. Should that code ever become accessible, it represents a huge security risk when endpoints and access credentials are exposed.

Security pitfalls should always be avoided, and they don't always come from the internet. Depending on your programming language, you may have to watch out for buffer overflows, cross-site scripting, and similar problems. Learn about the possible risks inherent in your language and in hard-coding, and avoid them as much as possible. Safety is not only a coding best practice but a best practice for every step of the project.

5. AVOID CODING WHEN TIRED

But working more doesn't necessarily lead to better code. Certain studies have shown that individuals are two to five times more likely to make a mistake when they are tired than when they are alert and full of energy. When you are tired or distracted, you'll inevitably make mistakes which either you'll have to fix later, making that time you spent essentially wasted, or you won't catch the mistake at all, meaning that the mistake will be found later in QA or maybe not until the end-user gets the product in their hand.

The link to it:

https://www.sleeek.io/blog/15-coding-best-practices-to-follow