

Coding Standards

Software engineering teams aim to build high quality and performance software using fast, efficient processes that can be repeated easily. However, that goal becomes much harder to achieve when teams face tight budgets, looming deadlines, and increasing customer expectations. Teams strive to write clean and reliable code that's easy to update and maintain. Yet again, that's not always easy to achieve. According to a report from CISQ estimated that poor-quality software cost the U.S. at least \$2.41 trillion in 2022 alone.

To help navigate this issue, developers will need to follow certain rules and best practices that help them write better code on a regular basis. These rules and techniques are known as coding standards. This paper will delve into what those coding standards are.

Coding standards, which are also called coding guidelines, are a set of rules that developers follow when writing code. These rules help keep the code consistent, organized, and easy to read, which makes it much easier for teams to work together on a project over time. Teams use coding standards to make it easier to maintain. Each team should clearly define its own standards based on the needs of the project, the type of work the company does, and the programming languages being used.

When coding standards are well set up and followed, even new developers can quickly understand and work within the code base. Ideally, the entire codebase should feel like it was written by one person and not a mixture of different styles.

Open coding standards are a set of rules that anyone can view or contribute to. These standards are shared publicly, which allows developers and security experts from around the world to help improve them. One example is the CERT C and C++ coding standard; this is published online as a wiki and updated through the input of the community.

On the other hand, a closed coding standard is private and only used within a specific company. These rules are not shared with the public and are meant only for the company. For example, a company will have its own internal guide that explains how to name variables, write comments, and follow certain patterns specific to its own software.

The biggest difference between open and closed standards is how often they change.

Open standards are very flexible and will update frequently to keep up with new practices.

Where closed standards usually change slowly and are updated only when the company's internal tools, needs, and workflows need to be updated.

Some things to keep in mind when deciding between open and closed ones for your organizations are:

- Maintenance – Because open standards change frequently, this can lead to inconsistencies over time. Closed coding standards will change less frequently and will be tailored to your team's tools and goals.
- Team Onboarding – Open standards will make it easier for new developers to get started quicker. Closed standards can slow the onboarding process because new hires will need time to learn your company's internal rules.

- Security – Closed standards will offer better protection by keeping internal coding rules private. This is best used for companies with critically sensitive information such as banks or medical applications.
- Time – Building your own coding standards will take time and expertise. This can be harder for smaller teams to do. Open standards are already made and maintained by the community.

In general, the closed standards would be ideal for enterprise or high security projects, while open standards work well for open-source or non-sensitive software.

Coding conventions are language specific guidelines that suggest how to write clean and well-organized code. They will mainly focus on the style and structure of the code, which helps developers keep their code consistent while still having the benefit of flexibility.

Different programming languages and developer communities often have their own unique conventions. After your team decides which coding conventions matter the most and officially adopts them, they will become part of your overall coding standards. This is what sets coding conventions apart. They are guidelines until your team turns them into required practices.

There are several best practices that help make code better. Below are some of the most important ones:

- Use Industry-specific Coding Standards – Coding standards can vary greatly depending on the industry you work in.
- Prioritize Code Readability – Readable code is easier to understand and maintain.

- Use Meaningful Names – Choose names that clearly describe what a variable, function, or class does. Good naming makes your code self-explanatory.
- Avoid Reusing Names for Multiple Purposes – Each variable should have a clear and specific name that reflects its role in the code. Additionally, avoid using the same variable or function name for different tasks as this can be confusing.
- Add Comments and Prioritize Documentation – Comments are essential for explaining what the code is doing, especially in complex sections. Good comments will make your logic, algorithms, or business rules easier for others to understand.
- Efficient Data Processing – Break your code down into small, reusable functions or modules to make it easier to manage.
- Effective Version Control – Use version control tools like Git to manage code changes.
- Code Review – Involve QA teams during refactoring to catch potential issues early.
- Formalize Exception Handling – Exceptions are unexpected issues that disrupt the normal flow of a program, and if not addressed properly, they can cause crashes.
- Security and Privacy – Protect the user data by ensuring that all insights are gathered without violating privacy. Only use data that users consent to sharing.
- Standardize Module Headers – The use of consistent headers makes code easier to read and manage. Every module should follow the same format.
- Make Daily Backups a Habit – Data loss can happen at any time due to crashes, power failure, or hardware issues. To avoid setbacks, back up your code.

- Consider Open Vs. Closed Standards – When choosing coding standards, decide whether an open or closed approach is best for your project.

Coding standards are extremely important because they ensure that developers create high-quality code. But what does high-quality code mean? High-quality code ensures that your software is safe, secure, and reliable for you and your users. When your code is these three things, you and your team can expect to reap the associated benefits. (Codacy n.d.)

Coding standards provide clear rules to follow which in turn help your team write clean, efficient, and easy to understand code. When these standards are followed, even new developers can quickly understand and work with any part of the codebase. Additionally, it will reduce the complexity of the code. This means that with cleaner and more readable code, it will be easier to maintain and ultimately control bugs.

In summary, coding standards are simply the foundation of writing high quality code. As development teams face challenges, having these clear guidelines will make for cleaner and better code. These standards help reduce errors and make onboarding new developers smoother. By following coding standards and conventions, teams can ensure their code is easier to read. These best practices such as using meaningful names, organizing code into small functions, commenting effectively, applying version control, and handling exceptions properly all contribute to better software. Choosing the right approach, whether you choose open or closed, depends on the project's needs. No matter the path, strong coding standards will not only lead to better software, but also a more productive team.

References:

Consortium for Information & Software Quality. (2022). *The cost of poor software quality in the U.S.: A 2022 report*. Author. <https://www.it-cisq.org/wp-content/uploads/sites/6/2022/11/CPSQ-Report-Nov-22-2.pdf>

QA Systems GmbH. (n.d.). *CERT Coding Standard for C and C++*. QA Systems. Retrieved July 22, 2025, from <https://www.qa-systems.com/solutions/cert/>

BrowserStack. (2024, June 28). *Coding standards and best practices to follow*. BrowserStack Guides. <https://www.browserstack.com/guide/coding-standards-best-practices>

Codacy. (2025, March ?)*. *Coding standards: What are they and why are they important?* Codacy Blog. <https://blog.codacy.com/coding-standards>