**UNIT 10**

Code quality measures of how well can code communicate between developers. Similar software may have an evident difference in code from developer to developer, but what matters most is that the quality of code remains the same throughout. A quality code is one that is clear, simple, well tested, [**bug-free**](https://www.codegrip.tech/productivity/tips-to-write-bug-free-code-during-interview/?utm_source=website&utm_medium=blog&utm_campaign=why-is-code-quality-important), refactored, documented, and performant. But the primary measure of high-quality code in compliance with the specification depends on the needs of the company.

Code quality is not a new parameter for code, and in fact, it has been in talks since 1970. It has become a necessity now to focus more on the quality of code that around 90% of businesses use [**code review tools**](https://www.codegrip.tech/?utm_source=website&utm_medium=blog&utm_campaign=why-is-code-review-important) for improving code quality. Measuring the quality of code, on the other hand, can be complicated as parameters change in accordance with the company’s demand, so there is no one way to measure it. Key metrics to measure the quality of code are reliability, maintainability, testability, portability, and reusability. We would see below what importance these parameters play in developing/writing quality code.

**1)Explain Importance of Code Quality? [5 marks]**

**Making software robust**

End-users feel very uncomfortable if they receive a strange message when they do something wrong in software. Here is when robustness comes to play. The software can cope with errors during execution in spite of unusual conditions. High-quality software has clear and understandable error messages for end users.

**Increasing readability and ease of editing**

Good quality code makes sure that codes are written in such a way that makes them [highly readable](https://www.codegrip.tech/productivity/how-to-format-code-for-maximum-code-readability/?utm_source=website&utm_medium=blog&utm_campaign=why-is-code-quality-important?). The use of comments, proper indentation, clear notations, and simplicity in the flow are some factors. Editing code is also a more comfortable job with high-quality code as they are easy to read and implement changes. Communication of code is more straightforward and hence also promotes inter-team learning.

**Making program sustainable**

Software is said to be sustainable when it can survive over time with minimal changes. An average software life is accounted to be around six years, but poor software quality does not last even half of this time duration. It’s because, with the constant development of operating programs and hardware, change in software escalates. It’s challenging and sometimes impossible to perform transformations on the poor-quality of code.

**Promotes easy transferability**

As discussed above, high-quality code has high readability and is easy to edit. Another added importance is that code quality practices make the translation of software across platforms straightforward. With numerous platforms for software to work on, it is essential to have direct transferability with minimal changes required.

**Decreasing technical debt**

Software development is itself a high-budgeted job with expectations of software to work as long as it can, with minimum error. But software with poor quality is set to fail early unless a significant number of changes are brought into the program repeatedly and hence increasing the technical debt. The extra development work is time and capital-consuming, which a high-quality code avoids.

Badly written code could lead to financial losses or waste of time needed for further maintenance, enhancements, or adjustments to software. You write your code once, but then follow it numerous times after. Hence, documenting your code becomes really important, and naming conventions become really important.

**2)List of Top Code Quality Tools [3 marks]**

* PVS-Studio.
* SonarQube.
* Crucible.
* Codacy.
* Upsource.
* Review board.
* Phabricator.
* Deepscan.

**3)What does container mean in software? [2 marks]**

Containers are **packages of software that contain all of the necessary elements to run in any environment**. In this way, containers virtualize the operating system and run anywhere, from a private data center to the public cloud or even on a developer's personal laptop.

 Containerization is **the packaging together of software code with all it's necessary components like libraries, frameworks, and other dependencies so that they are isolated in their own "container."**

**4) Explain the Benefits of containerization? [3 marks]**

* Portability.
* Efficiency.
* Agility.
* Faster delivery.
* Improved security.
* Faster app startup.
* Easier management.
* Flexibility.