

Static Code Analysis: What, When, Why, Why Not, & How.

Errors- A word that is not so peaceful in the world of developers and software development. So, without any doubt, developers work hard to overcome these errors as well as find methods of eliminating them even before they occur. We are here to help you regards one of those methods. (That's great, No?). So, as promised in the title, this guide will cover every significant perspective regards the aspect of Static Code Analysis.

Let's start!

What?

Starting with the answer to 'What?' is always the best place to dive into the explanation. So, what is static analysis?

Static means 'non-executing', and analysis means 'examination'. Static analysis is the process of finding and fixing the errors in source code before running it.

This process of static analysis was majorly dependant on human-skills, but presently, one can find several tools that automate this process. Many compilers come in hand with the integrated static analysis tools, making it easier for you.

When?

Next comes, when can you implement the process or method of static analysis.

The static analyzation of code takes place before the testing phases. The testing here signifies the unit testing. It allows you to have instant feedback on the code, leading to error solving in time, which prevents them from incrementing to the huge ones.

Why?

Knowing the purpose of any method or process is a significant aspect. Following are ways of how static code analysis can assist you in your development process:

1. It allows you to find and fix the programming errors in your source code. The unusual aspect about the static analysis is that it helps you spot the faults along with the exact position of it relevant to the code.
2. The technique allows you to not make any transgressions from the syntax of that particular programming language.

3. The static analysis explores and detects all sort of variables which are undefined.
4. A unique aspect of static analysis is that it also assists you in pointing out the vulnerabilities in your code concerning the security of data.
5. Lastly, it allows you to ensure to not make any violation of the coding standards.
6. This method gives you instant feedback which is always a helpful aspect when it comes to coding.
7. Static Code Analysis is a scalable method which permits you to make use of it on several source codes at a time.
8. The technique proves to be useful even in offline-based development environments.

Why Not?

Well, static code analysis plays a significant role in the development environment, but it also holds some shortcomings. The shortcomings of static code analysis include:

1. Method of Static Analysis may not detect all the possible set of faults or vulnerabilities in your source code with a hundred per cent guarantee.
2. It does not assist you in external parts of your development process that may include setting up user guides, code description and the style of code. Also, it may not ensure the detection of the violation of all the coding standards.
3. Static code analysis does not help you with the detection of logical errors in your code.
4. The technique does not handle the analyzation of additional libraries.
5. Static analysis may detect some false positives (Detecting the parts of code as weak or as faulty which are not)

All in all, there are few shortcomings to this method, but the fact that the benefits of it still outweigh the demerits stands high.

How?

Time to answer the 'How?' now

❖ Process of static analysis

The process of static analysis is simple yet significant. It's a simple four-step process:

Write	Execute	Resolve	Repeat
Write your source code.	Execute the static code analysis tool you are using. It will detect the errors or vulnerabilities in your code.	Resolve those errors according to the syntax and standards.	Repeat. (Well, development is always a continuous process).

❖ **Process of choosing the right static analysis tool:**

Another significant point here is how to choose the right analysis tool for your source code? Take a look at the following three questions, any analysis tool that satisfies the three of them is the one for you:

1. Does it provide analysis on the programming language being used by you?
2. Does the cost of the tool fit in your budget?
3. Does this tool abide by the coding standards of your organization?

There can be some additional questions like; *Does it depicts your code health quantitatively? Does it allow the integration with platforms like GitHub, Bitbucket or GitLab?* etc. But these all depend on the detailed specification of your code.

❖ **Categories of static analysis:**

As a developer, you should know the types of static analysis, which are as follows:

1. **Analysis of Data:** The method of assessing the behaviour of data in the dynamic state though it remains in static one.
2. **Analysis of Interface:** It deals with approaching the source code concerning the standards of making it function effectively for the required interface.
3. **Analysis of Faults:** The process of detecting the faults or failures in the components of a model.
4. **Analysis of Control:** It deals with the aspect of ensuring the proper flow of data in control structures.

Bonus Point: Why Us?

Till now, you would have for sure comprehended the significance of static code analysis in the development process.

We, at Codacy.com, provide you with the complete static analyzation of code. Let's answer some of your questions right away:

- *Do you provide code analysis for my programming language?*

No matter which programming language you are using to write your code, we are here for you, as our tool covers the code analysis of up to 30 programming languages.

- *Does the code analyzation by Codacy abide by the coding standards?*

Our team got this! Don't worry. We make sure to provide you with the analysis as per the present and accepted coding standards.

- *Does Codacy provide integration with Git providers?*

Well, yes! We can review any pull request and commit from GitHub, GitLab and Bitbucket to provide you with the quality checks.

- *But, does it fit my budget?*

We provide you with a ton of features at a very nominal cost. All you have to do is to reach out to us, and our team will handle the rest.

- *Will it disturb my current workflow?*

No! We provide you with static code analyzation directly from your current workflow.

Find us [here](#). Our team is always on its toes to provide you with the best.