There are two major styles of testing that can performed to ensure the quality of the software: static & dynamic.

Static testing is a more proactive process for software testing that involves examining the code, its design documents, & other additional artifacts without running the code. This type of testing primarily consists of code reviews, walkthrough, & static code analysis to assist in identifying errors & areas of potential improvement during the early stages. Static testing helps the tester focus on preventing defects before rollout, adhering to coding standards, & verification that the design begins built aligns with the project requirements before the software is executed & released.

On the opposite end, we have dynamic testing, which involves execution of the software to evaluate its functionality, performance, & behavior in different scenarios. This method relies upon unit testing, integration testing, system testing, & acceptance testing. The wide variety of testing methods allows the testers to view different results, with intent to focus on identifying runtime defects that may only be visible when the program is happening. Intermittent issues like this can completely break down the reliability of a program if roll out and not identified. Dynamic testing provides additional validation that the software will behave as intended for a real-world use cases & scenarios, ensuring its functionality & end user satisfaction.

Some noticeable differences between these two testing methods can be viewed by their different approaches & timing. Static testing is viewed for its efficiency in detecting errors during development process, target documentation, design phases, & code structure changes. This is a proactive effort to address potential efforts. Dynamic testing is more reactive, as it occurs after the code execution is complete. This is followed by efforts to correct runtime problems & make it functional.

By using both methods of testing, testers can provide crucial feedback & comprehensive software quality assurance & control. Static testing facilitates early defect detection, reducing the time and cost of addressing issues later in the development cycle. It also reinforces compliance with coding standards and best practices. Meanwhile, dynamic testing ensures that the software's actual performance, usability, and functionality meet expectations. Integration of both approaches allows for well-rounded testing to address errors in all stages, leading to more reliable, efficient, & user-friendly software for distribution.

GeeksforGeeks. (2024, July 12). *Difference between static and dynamic testing*. GeeksforGeeks. <https://www.geeksforgeeks.org/difference-between-static-and-dynamic-testing/>

Hamilton, T. (2024, July 5). *Static vs dynamic testing: Difference between them*. Guru99. <https://www.guru99.com/static-dynamic-testing.html>

Brian Hambling, Peter Morgan, Angelina Samaroo, Geoff Thompson, & Peter Williams. (2019). *Software testing: An ISTQB-BCS certified tester foundation guide - 4th edition* (4th ed.). British Computer Society.