**Exploring Static and Dynamic Testing Techniques**

**Introduction**

Testing is a big part of software development. It helps make sure the software works correctly and is secure. Two main types of testing are used—static testing and dynamic testing. Each one has its own purpose, and both are important for making sure software is reliable. This journal explains what static and dynamic testing are, how they’re different, and why using both types is valuable.

**What is Static Testing?**

Static testing is a way to check code and documents without actually running the program. This type of testing involves reading through code, inspecting it, and doing reviews to find any issues early on. Static testing is useful because it can catch errors before the software even starts running, which can save a lot of time and effort later.

**What is Dynamic Testing?**

Dynamic testing, on the other hand, is about running the software to see how it performs. This involves tests like unit testing (checking small parts of the code) and user acceptance testing (making sure it meets user needs). Dynamic testing shows if the software works correctly and handles different situations as expected.

**Differences Between Static and Dynamic Testing**

1. **Running the Code**: Static testing doesn’t involve running the software, while dynamic testing does.
2. **Purpose**: Static testing looks for issues in the structure and design of the code. Dynamic testing checks if the software works as it should.
3. **Timing**: Static testing happens early in the process, often before the code is finished. Dynamic testing usually happens later, once the code is ready to run.

**Why It’s Important to Use Both Static and Dynamic Testing**

Using both types of testing gives a more complete check of the software. Static testing catches issues early, which saves time and money by avoiding major problems down the line. Dynamic testing ensures the software actually works and meets the needs of users. By combining both approaches, developers create a balanced testing process that makes sure the software is both well-structured and functional.