UNIT 4

SYSTEM TESTING

Static and Dynamic Testing

As we know that testing is the most important stage in the process of delivery of any application or software. Testing not only validates the quality of an application but also provides an opportunity to the developer to improve its product.

Every application is being developed in some high or low level language which means some code has been written for its development so on the basis of execution of code written for the application there is classification of testing namely **static testing** and **dynamic testing**.

In this article, we will discuss all the important differences between static testing and dynamic testing. Let's start with some basics of static testing and dynamic testing.

What is Static Testing?

Static testing is the testing in which code written for application is not executed during testing phase and only review of code is performed and basis on which defects and code quality has been examined. As its name implies, static testing performs the static verification of the code. It targets the assessment of program code and documentation.

Static testing is generally performed before the compilation of the code. Static testing involves two types of testing techniques namely review and static analysis.

What is Dynamic Testing?

Dynamic testing there is execution of code written for the application and then defects and application behavior has been examined. Dynamic testing is performed to examine the behavior of the software based on dynamic inputs. The dynamic testing of a software product is performed after compilation of the software code.

MC4102-OOSE

Dynamic testing is classified into two types namely – white box testing and black testing. In software testing techniques, the dynamic testing is one of the essential tools for detecting any security threats. The dynamic testing increases quality of the product.

Difference between Static and Dynamic Testing

The following table highlights the major differences between Static Testing and Dynamic Testing −

Parameter	Static Testing	Dynamic Testing
Definition	Static testing is the testing in which code written for application is not executed during testing phase and only review of code is performed and basis on which defects and code quality has been examined.	Dynamic testing there is execution of code written for the application and then defects and application behavior has been examined.
Nature of testing	As name states static testing does the static verification process in which the requirement and corresponding written code has been verified.	Dynamic testing does the validation process which examines the expected behavior of the application based on dynamic inputs provided to the application.
Testing target	Static testing targets to the assessment of code and documentation.	Dynamic testing targets the runtime bugs/bottlenecks in the software system.
Prerequisite	For static testing, a checklist of application process and documentation is required.	For dynamic testing, test cases for execution are to be developed.
Stage of testing	Static testing generally get performed before compilation of code	Dynamic testing mostly performed after compilation of code.
Cost to Company	In Static testing, the cost of finding the defects and fixing them is less. Also, the Return on Investment is high because static testing is carried out at an early stage of development.	In case of Dynamic testing, the cost of finding and fixing the defects is high. Also the Return on Investment (RoI) is low because this process is carried out after the development phase.

Conclusion

The most important difference between static and dynamic testing is that the static testing checks the defects in software without actual execution of the software code and it analyzes the static behavior of the software, while dynamic testing is used to analyze the dynamic behavior of the software.

Since testing is of two types like 1) Static testing 2) Dynamic testing; accordingly the tools used during these testing are also known as

1) Static testing tools

2) Dynamic testing tools

Static testing tools seek to support the static testing process whereas dynamic testing tools support dynamic testing process. It may be noted that static testing is different from dynamic testing.

Few points of differences among static and dynamic testing are as under:

	Static Testing	Dynamic Testing
1	Static testing does not require the actual execution of software.	Dynamic testing involves testing the software by actually executing it.
2	It is more cost effective.	It is less cost effective.
3	It may achieve 100% statement coverage in relatively short time.	It achieves less than 50% statement coverage because it finds bugs only in part of codes those are actually executed.
4	It usually takes shorter time.	It may involve running several test cases, each of which may take longer then compilation.
5	It may uncover variety of bugs.	It uncovers limited type of bugs that are explorable through execution.
6	It can be done before compilation.	It can take place only after executables are ready

Software testing tools are frequently used to ensure consistency, thoroughness and efficiency in testing software products and to fulfil the requirements of planned testing activities. These tools may facilitate unit (module) testing and subsequent integration testing (e.g., drivers and stubs) as well as commercial software testing tools.

Testing tools can be classified into following two categories:

Static Test Tools: These tools do not involve actual input and output. Rather, they take a symbolic approach to testing, i.e. they do not test the actual execution of the software. These tools include the following: ,

1) Flow analyzers: They ensure consistency in data flow from input to output.

2) Path tests: They find unused code and code with contradictions.

3) Coverage analyzers: It ensures that all logic paths are tested.

4) Interface analyzers: It examines the effects of passing variables and data between

modules.

Dynamic Test Tools: These tools test the software system with 'live' data. Dynamic test tools include the following

1) Test driver: It inputs data into a module-under-test (MUT).

2) Test beds: It simultaneously displays source code along with the program under execution.

3) Emulators: The response facilities are used to emulate parts of the system not yet developed.

4) Mutation analyzers: The errors are deliberately 'fed' into the code in order to test fault tolerance of the system.