

SOFTWARE QUALITY ASSURANCE

The major key areas of SQA are

- Bugs and Debugging
- Testing strategies.
- The impact of an object orientation on testing.
- How to develop test cases.
- How to develop test plans.

Two issues in software quality are:

- Validation or user satisfaction
- Verification or quality assurance.

Elimination of the syntactical bug is the process of debugging. Detection and elimination of the logical bug is the process of testing.

Error Types:

- Language errors or syntax errors
- Run-time errors
- Logic errors

Identifying Bugs and Debugging

- The first step in debugging is recognizing that a bug exists.
- Sometimes it's obvious; the first time you run the application, it shows itself.
- Other bugs might not surface until a method receives a certain value, or until you take a closer look at the output

However, these steps might help:

- Selecting appropriate testing strategies
- Developing test cases and sound test plan.

Debugging Tools

- Debugging tools are a way of looking inside the program to help us determine what happens and why.

- It basically gives us a snapshot of the current state of the program.

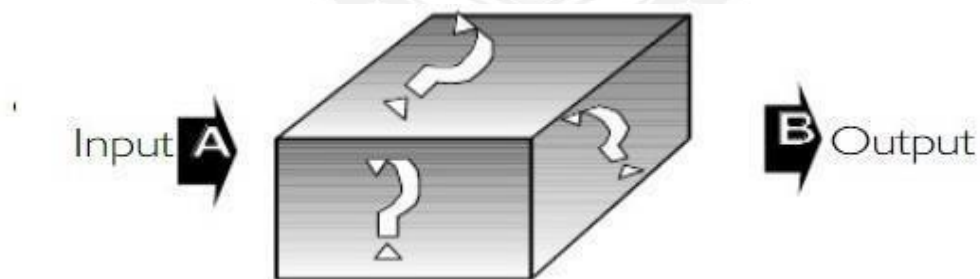
Testing Strategies

There are four types of testing strategies, These are:

- ✓ Black Box Testing
- ✓ White Box Testing
- ✓ Top-down Testing
- ✓ Bottom-up Testing

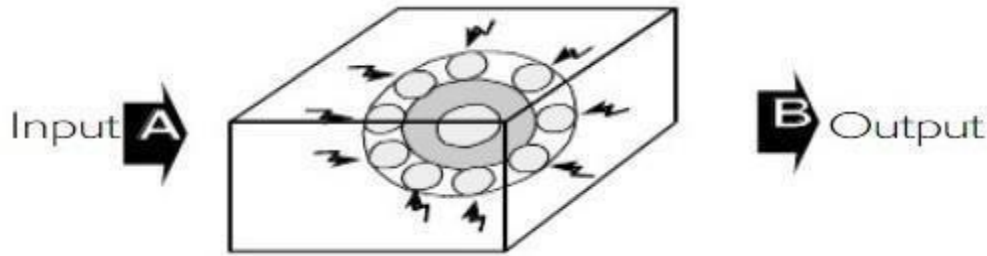
Black Box Testing

- In a black box, the test item is treated as "black" whose logic is unknown.
- All that's known is what goes in and what comes out, the input and output
- Black box test works very nicely in testing objects in an Object-Oriented environment.



White Box Testing

White box testing assumes that specific logic is important, and must be tested to guarantee system's proper functioning. This testing looks for bugs that have a low probability of execution that has been overlooked in previous investigations. The main use of this testing is error-based testing, when all level based objects are tested carefully.



One form of white box testing is called path testing

- It makes certain that each path in a program is executed at least once during testing.

Two types of path testing are:

- Statement testing coverage- The main idea of the statement testing coverage is to test every statement in the objects method executing it at least once.

Branch testing coverage –The main idea here is to perform enough tests to ensure that every branch alternative has been executed at least once under some test. It is feasible to fully test any program of considerable size.

Top-down Testing

It assumes that the main logic of the application needs more testing than supporting logic.

Bottom-up Approach

- It takes an opposite approach.
- It assumes that individual programs and modules are fully developed as standalone processes.
- These modules are tested individually, and then combined for integration testing.

System Usability & Measuring User Satisfaction

- Verification

- "Am I building the product right?"

Validation

- "Am I building the right product?"

Two main issues in software quality are
Validation or user satisfaction and
verification or quality assurance.

•The process of designing view layer classes consists of the following steps:

1. In the macro-level user interface (UI) design process, identify view layer objects.
2. In the micro-level UI, apply design rules and GUI guidelines.
3. Test usability and user satisfaction.
4. Refine and iterate the design.

Usability and User Satisfaction Testing

Two issues will be discussed:

1. Usability Testing and how to develop a plan for usability testing.
2. User Satisfaction Test and guidelines for developing a plan for user satisfaction testing.

•The International Organization for Standardization (ISO) defines usability as the effectiveness, efficiency, and satisfaction with which a specified set of users can achieve a specified set of tasks in particular environments.

• Defining tasks. What are the tasks?

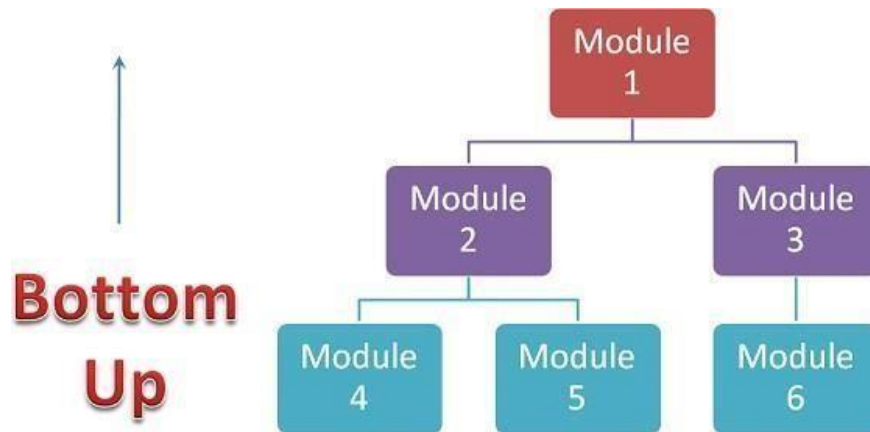
Defining users. Who are the users?

• A means for measuring effectiveness, efficiency, and satisfaction

The phrase two sides of the same coin is helpful for describing the relationship between the Usability and functionality of a system.

Bottom – Up Testing

It supports testing user interface and system integration. In the bottom-up strategy, each module at lower levels is tested with higher modules until all modules are tested. It takes help of Drivers for testing

**Advantages:**

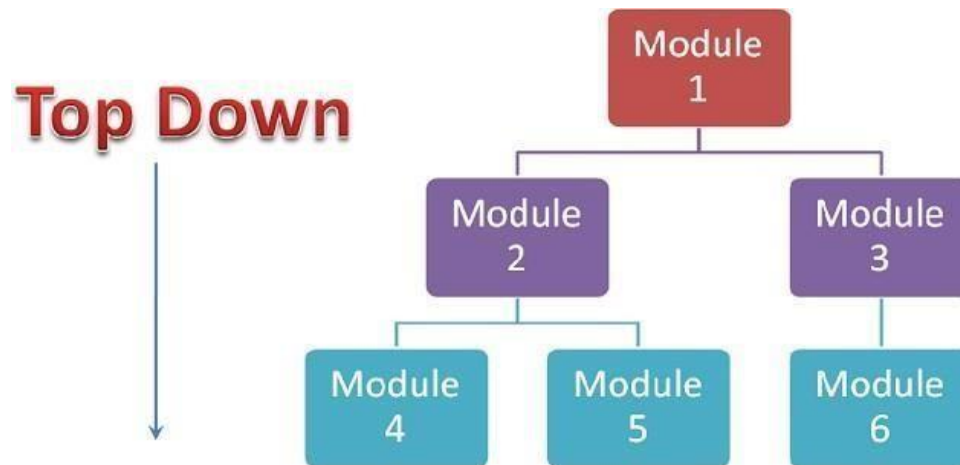
- Fault localization is easier.
- No time is wasted waiting for all modules to be developed unlike Big-bang approach

Disadvantages:

- Critical modules (at the top level of software architecture) which control the flow of application are tested last and may be prone to defects.
- An early prototype is not possible

Top-down Testing:

In Top to down approach, testing takes place from top to down following the control flow of the software system. Takes help of stubs for testing. It starts with the details of the system and proceeds to higher levels by a progressive aggregation of details until they fit requirements of system.



Advantages:

- Fault Localization is easier.
- Possibility to obtain an early prototype.
- Critical Modules are tested on priority; major design flaws could be found and fixed first.

Disadvantages:

- Needs many Stubs.
- Modules at a lower level are tested inadequately.