Software Testing

Software Testing is the process of executing a program or system with the intent of finding errors

Definition: Testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item.

Necessary Of Software Testing

- Its necessary because of human errors, some unimportant functions and mistakes
- We need to check everything and anything we produce because things can go wrong
- To point out the defects and errors that were made during the development phases.
- Its required for an effective performance of software application or product.

Testing software is operating the software under controlled conditions to:-

- Verification
- Validation
- Error detection

Testing Fundamentals

Testing is the phase where errors from the earlier phases must be detected.

The success of testing for errors in program depends on the test cases.

Testing is the process of providing the program with test inputs.

- Error : An error is a mistake or misunderstanding on the part of a software developer.
- In the category of developer, we include software engineers, programmers, analysts and testers.
- There are 3 types of errors:

Syntactical Error: Its deviation from the syntax of program

logical error: Its deviation from logic of the program.

Executional error: This occurs during exection of the program.

- Failures: A failure is the inability of a software system or component to perform its required functions within specified performance requirement.

Eg: Break fail $\rightarrow$ Car accident

- Faults : A faults (defect) is introduced in to software as the result of an error. Its also called as “BUGS”.

it’s the condition that causes the software to fail to perform its required function.

Eg: ATM not functioning.

Differentiate Between Failures and Faults

Failure	Fault
* Its External results of the program Operation from the requirements.	* when executed under particular Condition causes of failure.
* Failure is something Dynamic.	* A fault is a property of the program rather than a property of its Execution or behavior.
* Failure can be defined as Deficiency in performances, attributes and response time.	
* Various failure classes are : Recoverable, unrecoverable, non corrupting, and corrupting.	* Various fault classes are : data fault control fault, I/O faults, storage mgt Faults, Exception mgt faults

Verification and validation

Verification	validation
*The process of evaluating work-product Of a development phase to determine Whether they meet the specified Requirement For that phase.	The process of evaluating s/w during or at the end of the development process to determine whether it satisfied business requirement.
* Are we building the product Right?	Are we building the right product?
*Plans,Requirement,Design,code,testcase	The actual product /software
*Activities: Reviews and inspections	Testing.

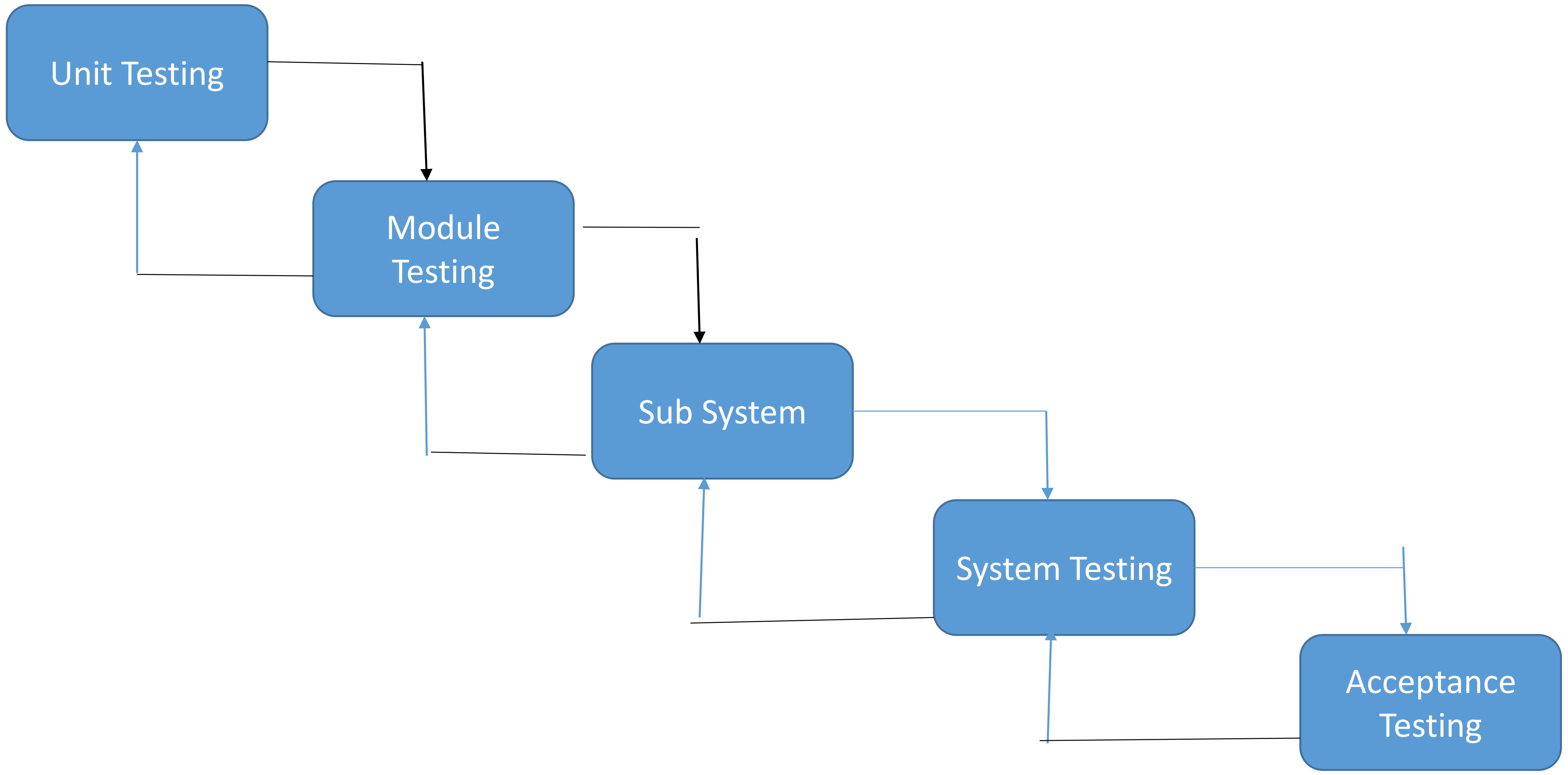
TEST PROCESS

- The general test process is the creation of test strategy (execution of test cases).
- Creation of a test plan/design and execution of test.
- Test data are input that have been devised to test the system.

The stages In testing process are:

1) Unit Testing
2) Module testing
3) Sub-system testing
4) System testing
5) Acceptance testing

Test Process



1) Unit Testing: individual components are tested to ensure that they operate correctly. Its tested independently without referring other system components.

2) Module Testing: it’s a collection of dependent components such as object class, an abstract data type. A module encapsulates related components so can be tested without referring other system modules.

3) Sub- system Testing: Testing collection of modules which have been integrated into sub system. problems arise in large software system are sub system interface mismatches.

4) System Testing: testing process is concerned with finding errors that result from unanticipated interactions between sub system and system components.

Its also concerned with validating that the system meets.

5) Acceptance Testing: it's a final stage of the testing process, before accepted by the customer. Tested by the user to check it satisfies the requirement.

What is Test Case:?

Test case is a scenario made up of a sequence of steps and conditions or variables, where test inputs are provided and the program is run using those input and check the performs of it.

An expected result is outlined and the actual result is compared to it.

Certain working conditions are also present in the Test Case, to see How the program handles the condition.

A formal Test case can be divided into 3 main parts:

1) Information: it consists of general information about the test case such as case identifier, case creator info, test case version
2) Activities: it consists of the actual test case activities such as environment that should exist during testing, activities to be done at the initialization of the test
3) Results : are the outcome of a performed test case. Result data consists of information about expected result.

TEST CASE FORMAT

2 sample formats for writing Test case are:

1) Detailed
2) Simple

Test case ID	Purpose	Test created by	Test Environment	Prerequisites	Test Procedure	Test date	Expected Result	Actual result	comments
Serial no assigned to test case	Brief idea about case	Name of test creator	s/w or h/w which the test case is executed	Conditions that should be fulfilled B4 the test is performed	Steps to be performed in test	Input variables and data	What the program should do	What is actually done	Notes on the procedure

Detailed format

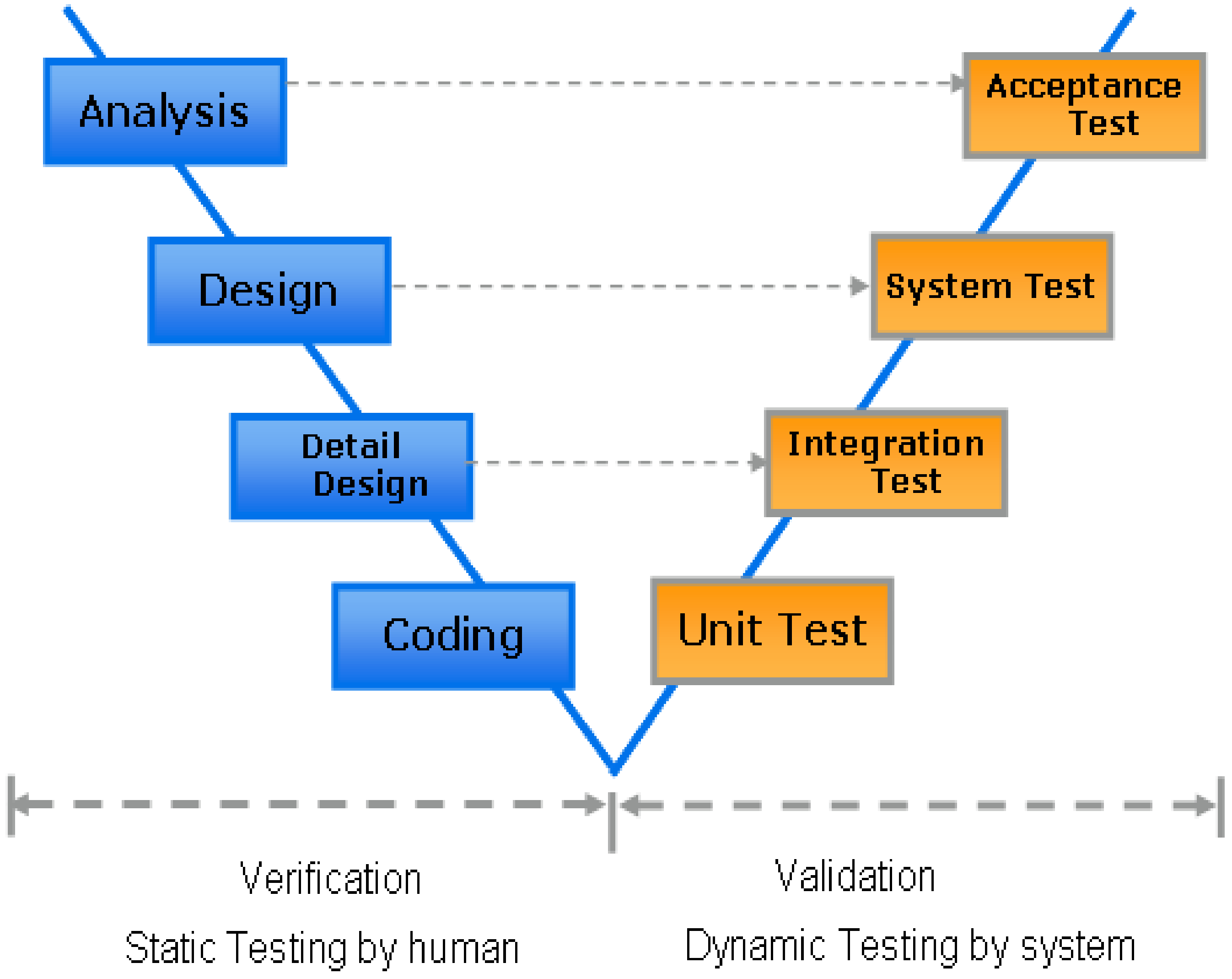
Simple Format:

Step No.	Serial no of step
Step or Activity	Detailed operation or procedure
Criteria for Success	Expected Results
status	Whether the code passed the test or not

Test model

V-Model

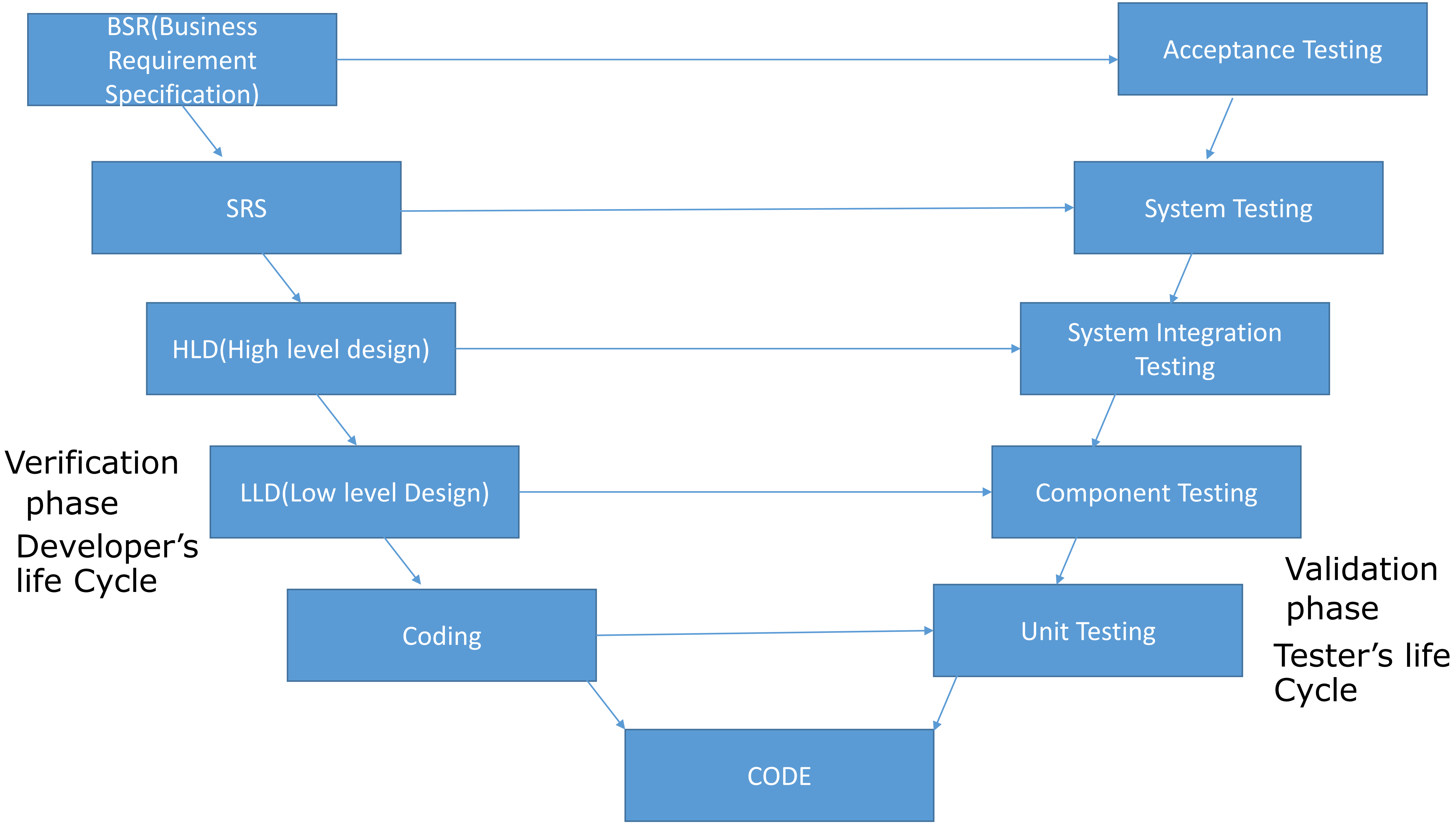
- Enhanced traditional waterfall model
- Verification Validation



When to use the v-model

It should be used for Small to medium sized projects where requirements are clearly defined and fixed.

The v-shaped model should be chosen when the technical resources are available with needed technical expertise.


BSR(Business Requirement Specification)
Acceptance Testing
SRS
System Testing
HLD(High level design)
System Integration Testing
LLD(Low level Design)
Component Testing
Coding
Unit Testing
CODE
Verification phase
Developer's life Cycle
Validation phase
Tester's life Cycle

1) Requirements BRS and SRS life cycle model just like water fall model, but this model is b4 development is started, a plan is created.

- Test plan focuses on meeting the functionality specified in the requirements gathering.

2) High- level Design(HLD) Focuses on System architecture and Design.

It provides overview of solution, platform, system, product and service/process

3) Low-Level Design(LLD) Phase is where the actual software components are designed. It defines the actual logic for each and every component of the system

4)Implementation where all coding takes place. Once the coding is complete, the path of execution continues up the right side of the V where the test plans developed earlier are now put to use.

5) Coding it’s the bottom of the V-shape model. Module design is converted into code by developers.

Advantages:

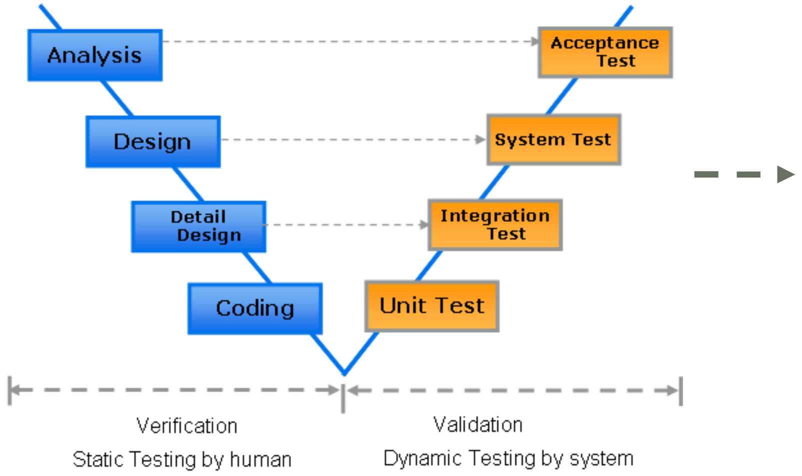
- Simple and easy to use
- Each phase has specific deliverables.
- Works well for where requirements are easily understood
- Higher chance of success.

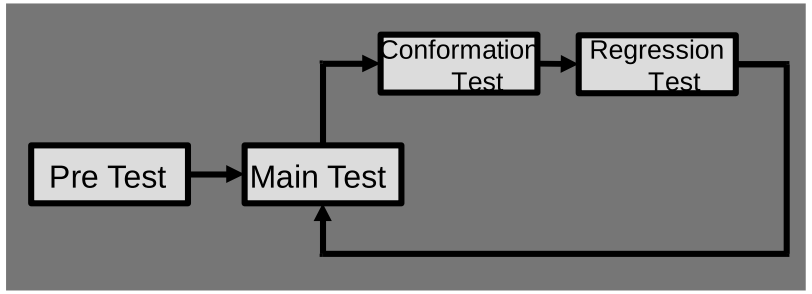
Disadvantages:

- Very inflexible like the waterfall model and adjusting is difficult and expensive
- Model doesn’t provide a clear path for problem found during testing phases.

Test Cycle

- Each test is consists of test cycle
- Test cycle
  - Pre test - check up testability
  - Main test
  - Conformation test - check up the defect found in previous "Main test"
  - Regression test - check up the impact from change





TEST CYCLE

Test Plan

- It’s a document which is created before the testing process.
- A document describing the scope, approach, resources and schedule of intended test activities.
- Test plans are not just management documents. They are also intended for software engineers involved in designing and carrying out system tests.

Test Plan Template

- The format and content of a software test plan vary depending on the processes, standards and test management tools being implemented.
- The IEEE standard for software test documentation, provides a summary of what a test plan should contain.

1) Test plan Identifier: Provide a unique identifier for document(configuration)
2) Introduction: Provide on overview of the test plan, specify the goals/objectives, specify any constraints.
3) References: List the related documents with the links, project plans.
4) Test Items: software products and there versions

5) Features to be tested: list the features of the software to be tested

6) Features not to be tested: list the features of the software not to be tested.

7) Item pass/fail criteria: specify the criteria that will be used to determine whether each test item has passed or failed testing.

8) Test Deliverables: Test plan
   Test cases
   Test Scripts
   Defect/ Enhancement logs
   Test report.

9) Test Environment: Specify the properties of test environment: h/w , s/w , n/w etc and list any testing and related tools.

10) Estimate: Provide a summary of test estimates and provides a link to the detailed estimation.

11) Schedule: Provide a summary of the schedule, specifying key test milestones, and provide link to the detailed schedule.

12) Staffing and Training needs: role and required skills

13) Responsibilities: team,role,individual

14) Risks: list the risk that have been identified

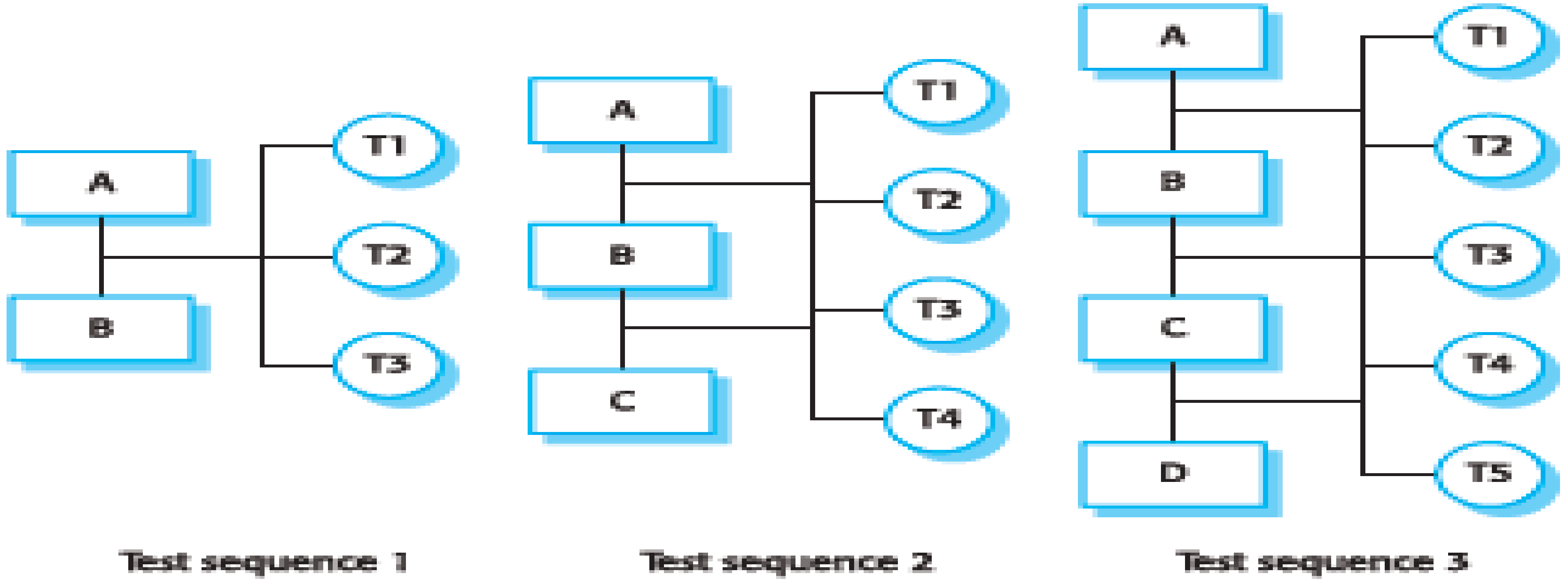
15) Approvals: specify the names and roles of all persons who must approve the plan.

Importance of Test Plan / Purpose of test Plan

1) It helps s/w engineers to think through the efforts needed for s/w product
2) It can and will help people outside the test group to understand why and how product validation takes place
3) Test plan document describes the objectives , scope, approach and focus of the s/w testing effort.
4) It includes test cases, conditions, the test environment, a list of related tasks, pass/fail.
5) Test plan should be documented, so that they are repeatable.

Test Strategy

1) Top- Down Testing
   - The main control module is used as a driver, and stubs are substituted for all modules directly subordinate to the main module.
   - Depending on the integration approach selected (depth or breadth first), subordinate stubs are replaced by modules one at a time.
   - Tests are run as each individual module is integrated.
   - On the successful completion of a set of tests, another stub is replaced with a real module
   - Regression testing is performed to ensure that errors have not developed as result of integrating new modules



Test T1, T2, T3 are first run on a system composed of module A and B.

Module C is Integrated and Test T1 T2 are repeated to ensure that there have not been unexpected interactions with A and B. Test T4 is also run on the system.

- Advantages:

No test drivers needed. (drivers are the dummy routines which are introduced that invoke a module)

Interface errors are discovered early

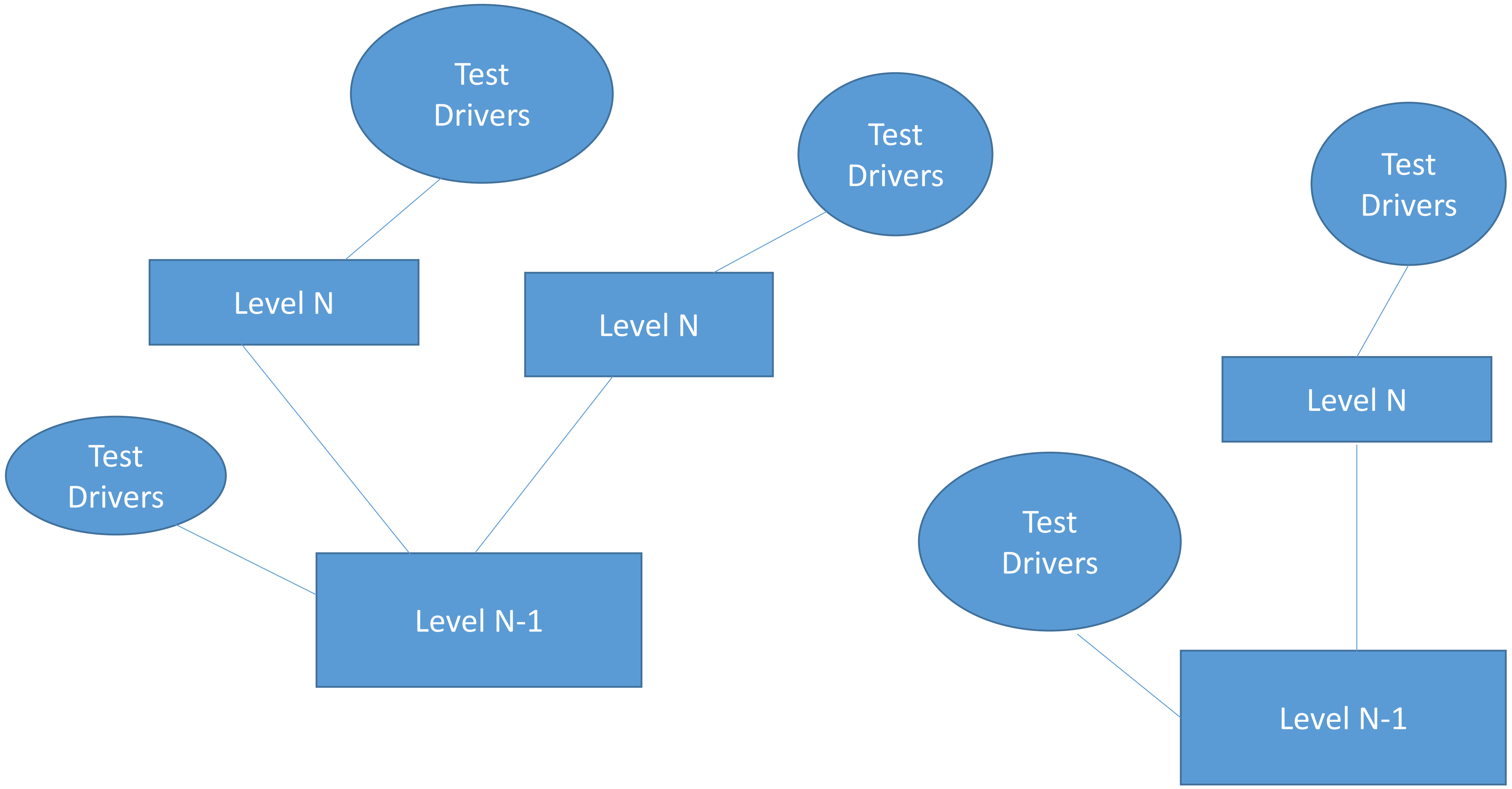
- Disadvantages:

Test stubs are needed.

Error in critical modules at low levels are found late.

2) Bottom-Up Integration

- Integration begins with the lowest-level modules, which are combined into clusters, or builds, that perform a specific software subfunction
- Drivers (control programs developed as stubs) are written to coordinate test case input and output
- The cluster is tested
- Drivers are removed and clusters are combined moving upward in the program structure


Test Drivers
Level N
Test Drivers
Level N
Test Drivers
Level N-1
Test Drivers
Level N
Test Drivers
Level N-1

- Advantages:

No test stubs are needed

Errors in critical modules are found early

- Disadvantages:

Test drivers are needed

Interface errors are discovered late.

3) Thread Testing

- Its normally applied for testing real time application program, its called event based system.
- Its an event based approach. Test are based on the events which trigger system action.
- it’s a testing strategy which may be used after processed or objects have been individually tested and integrated into sun system.

Definition: Thread Testing is a testing strategy which may be used after processed or objects have been individually tested and integrated into sub-systems.

4) Stress Testing

- Stress Testing is a form of testing that is used to determine the stability of a given system or entity.
- It involves testing beyond normal operational capacity
- stress Testing may have a more specific meaning in certain industries.

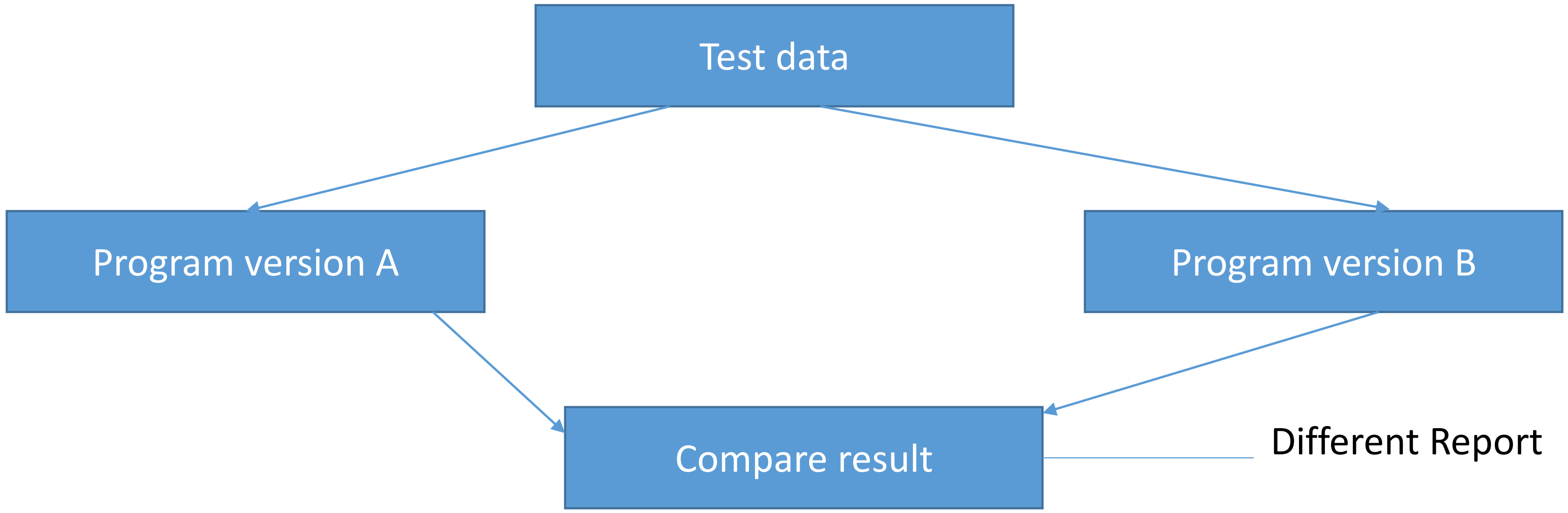
Eg: online Railway reservation system.

NOTE: stress systems are designed to handle a specified work.

Eg: OS may be designed to handle up to 200 separate terminal simultaneously

5) Back – to – back Testing

- It may be used when more than one version of a system is available for testing.
- Same test are presented to different versions of the system and outputs are compared.



Steps:

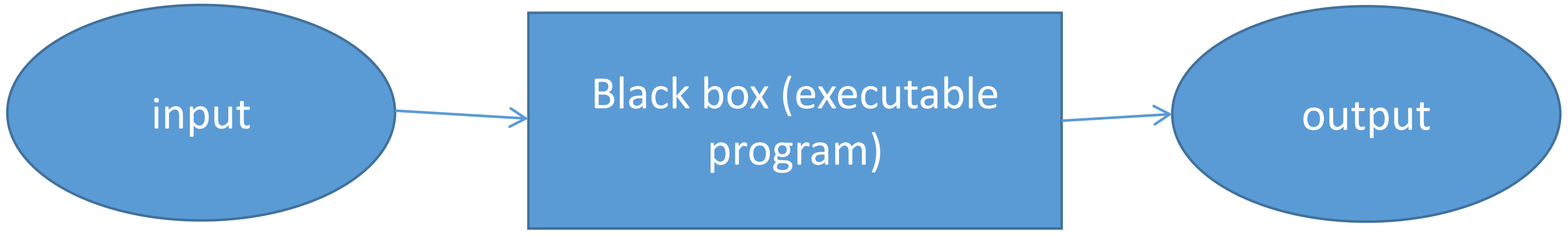
1) Prepare a set of test cases
2) Run one version of the program with these tests and save
3) Run another version with same test cases and save
4) Compare the result.

TYPES OF TESTING

1) Black box testing
2) White box testing
3) Alpha testing
4) Beta testing
5) Clean Room testing
6) Defect testing

BLACK BOX TESTING

- Black box testing is an approach to test where the program is considered as black box, whose behaviour can only be determine by studying its input and output.



- Program test case are based on the system specification.
- Problem : selection of inputs that have high probability of being member of the set.

Techniques used for selecting test case are

1) Equivalence partitioning
2) Boundary value analysis (BVA)

Equivalence partitioning, divides the input of program into classes of data from which data case are derived.

Input is divided into valid and in-valid

Boundary value analysis, every program has boundary value of input, many type times errors occurs at the boundary of equivalence class called boundary value analysis.

If the programmer specify wrong symbols:

$$\left.\begin{array}{l} < \text{ instead of } > \\ = \text{ instead of } \# \end{array}\right\} \quad \text{Leads to BVA}$$

Advantage

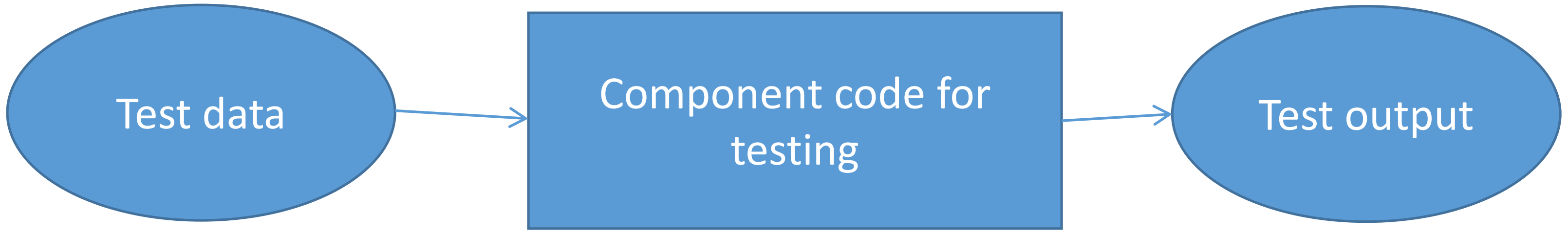
- Tests are done from user point of view
- Tester need not know programming languages or how software has been implemented.
- Test can be conducted independent from the developer.
- Test cases can be designed as soon as the specifications are complete
- Tester and programmer are independent of each other

Disadvantages

- Only small number of possible inputs can be tested
- Many program paths will be left untested.
- Tests can be redundant.
- Test case are hard to design
- Cannot be directed specific segments of code which may be very complex.

white box testing

- Its also called as Glass-Box testing or structural Testing or Clear box testing.
- Testing that takes place into account the internal mechanism of a system or component.



- Its used to test areas that cannot be reached from a black box testing.
- It requires programming skills to identify all paths through the software

Types of white box testing

1) Path testing
2) Condition Coverage
3) Statement Coverage
4) Branch Coverage
5) Data flow based testing
6) Mutation testing
7) Cyclomatic complexity

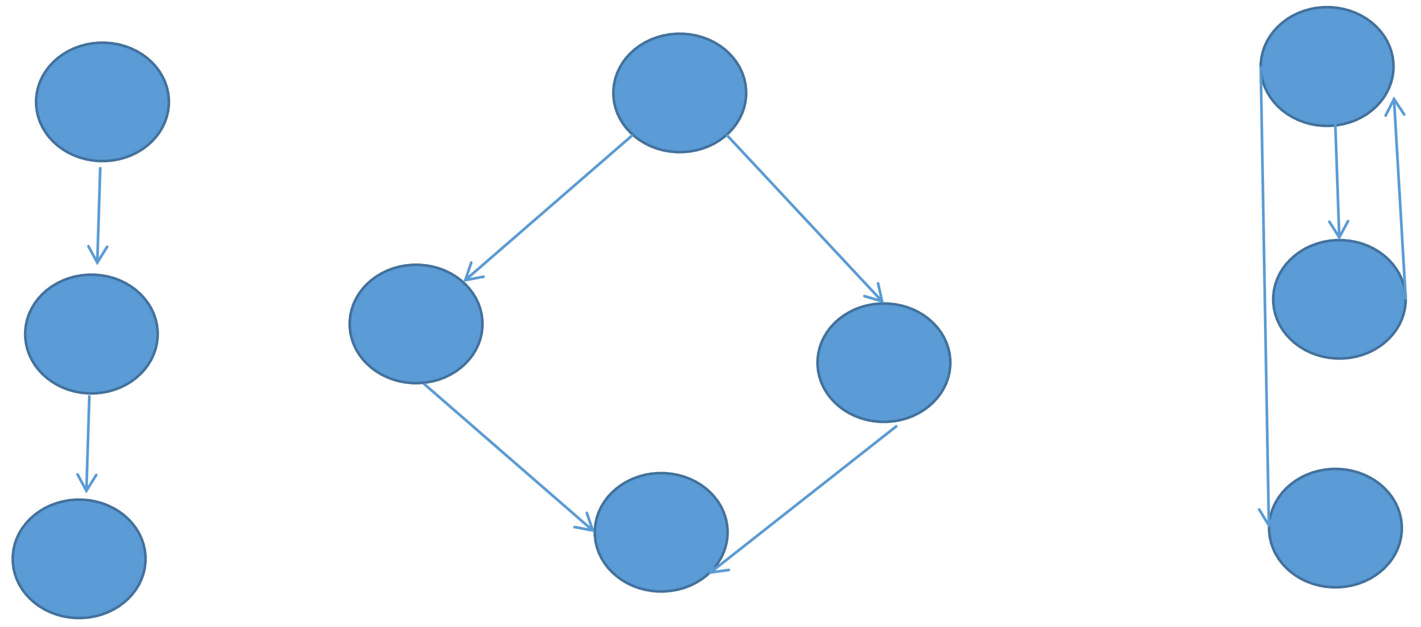
PATH TESTING

- Objective is to ensure that each path through the program is executed at least one
- Most coverage criteria are based on the number of statements, branches that are used by the test case.
- 2 major steps are:

Define all logical paths

Derive test case to execute all logical path

- Flow-Graph: the process of path testing starts with developing a graphical representation of logical control flow graph. Notations used in constructing a flow graph.



- A flow diagram consists of circles called NODES
- Representing decisions and arrows called Edges representing the flow of control

Cyclomatic Complexity

- The number of independent paths in a program can be found by computing the cyclomatic complexity (CC) of the program flow graph
- Its often referred as Mccabe’s Complexity
- **cc= Number of edges – Number of nodes +2**
- The minimum number of test cases needed to test all control statements equals cyclomatic complexity.
- All combinations of paths are necessarily executed.

Formula: $CC = E – N + 2$

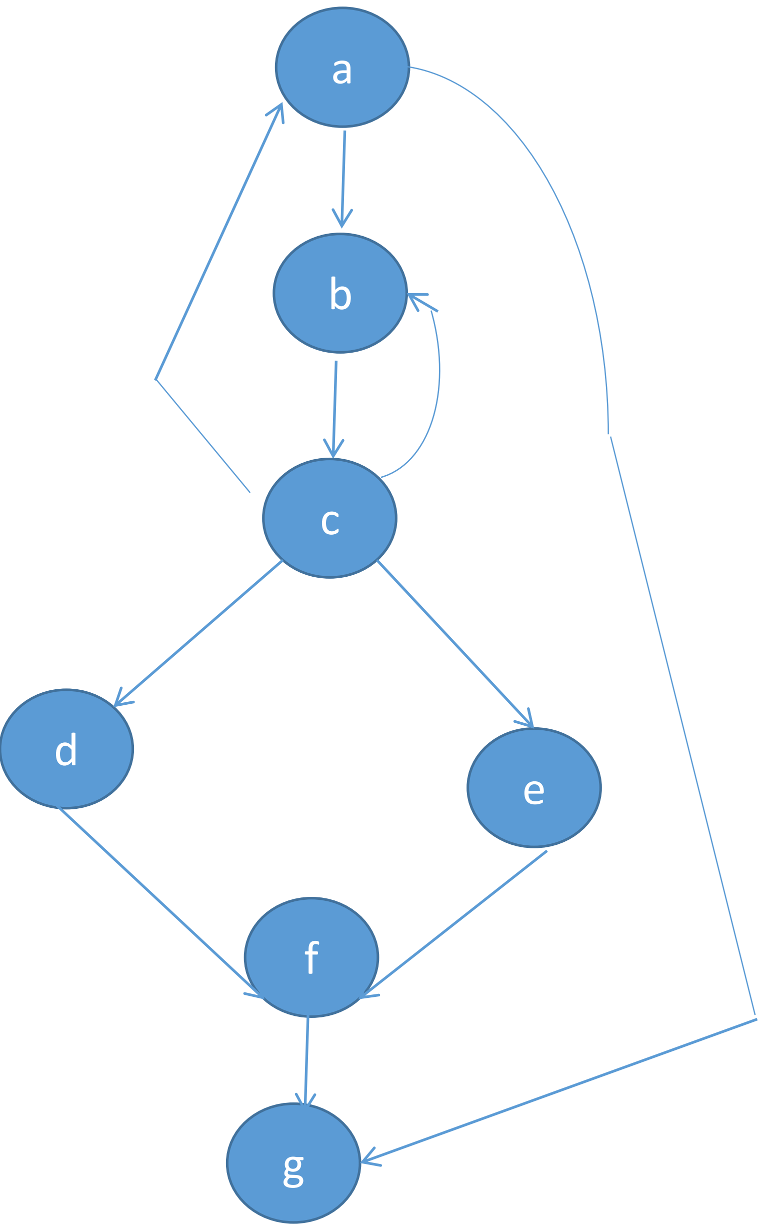
- Where

E --- the number of edges of the graph

N --- the number of nodes of the graph

FORMULA in simplified way it can be defined as:

$CC = D + 1$



no of edges (E) =

no of nodes (N) =

Hence Cyclomatic complexity is

cc [G] = E – N + 2

=

Advantages

- Testing can be commenced at the earlier stage. Need not wait for GUI to be available.
- Testing is more thorough with the possibility of covering most paths.

Disadvantages:

- Testing can be very complex.
- Highly skilled resources are required, with knowledge of programming and implementation.

Difference between black- and white-box testing

Black box	**white box**
1)Its the method in which the internal Structure of the item being tested in NOT known.	1)Its the method in which internal structure of item being tested is Known to the tester
2) We can do the Testing	2)we cant do the test performances of the application.
3) Requirement specification	3)Detail design

Black box	**white box**
4) It begin early in the s/w development I,e in requirement gathering phase itself	4) this approach has to wait for the designing has to complete.
5) Mainly applicable to higher level testing Acceptance and s/w testing testing	5) mainly applicable to lower levels of testing: unit and Integration

Alpha Testing:

- Its simulated for actual operational testing by user or customers.
- An independent test team at the developers site.

Beta Testing:

It comes after alpha testing. Versions of software known as beta version are released to limited users outside of the programming team.

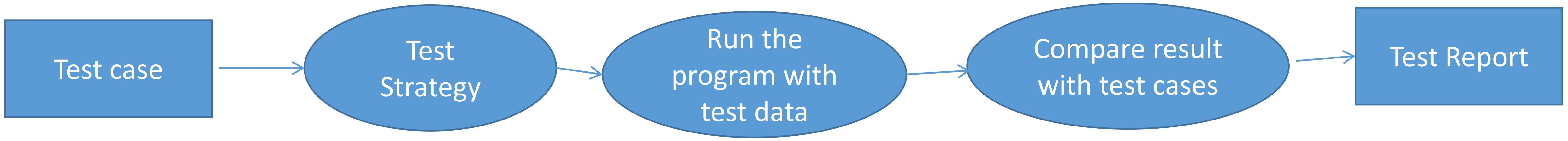
Clean Room testing:

Means defects are avoided by manufacturing in an ultra clean environment.

It makes use of incremental development model.

Defect Testing

- The goal of defect testing is to identify the defects before the system is delivered to the customer.
- A successful defect test causes failure in the system which is an indication of defects in the software.


Test case
Test Strategy
Run the program with test data
Compare result with test cases
Test Report

METRICS

$$\begin{tikzcd}
 & \text{Metrics} \arrow[dl] \arrow[dr] & \\
\text{Programmer Productivity} & & \text{Error removal efficiency}
\end{tikzcd}$$