Chapter-1

Software myths

① **Testing is a single phase in SDLC**

True :- (Reality) Testing start as soon as we get requirement specification for software. It continues in SDLC

② **Testing is easy**

Tester have to plan and develop mannually which requires complete understanding of project being developed with its overall design.

It is harder than development

③ **Software development is worth more than Testing.**

④ **Complete testing is possible**

not possible to provide all inputs to test software

⑤ **Testing starts after program development**

⑥ The purpose of testing is to check the functionality of software

The goal is to ensure the quality of software

⑦ Anyone can be tester

- understanding testing life cycle
- techniques to design test cases
- various tools and how to work on them

what is software testing ?

Short question → is the process of executing a program with intent of finding errors

definition → is a process that detects important bugs with the objective of having better quality software

Goals of software Testing

short-term

Immediate Goals

1. Bug discovery
2. Bug Prevention

Post Implementation Goals

Goals after product Released

1. Reduced maintenance cost
2. Improve software Testing process.

Long term Goals

- Reliability (1)
- Quality (2)
- customer satisfaction (3)
- Risk management (4)

Software Testing
↓ controlled by
Risk management

** Risk factorss
- cost
- time
- Resource
- Critical feature

Quality ↓ provides customer satisfaction

Chapter-2

Error :- Human mistake can occur due to developer's (1)

coding error

Why error occurs?

(II) developer misunderstood requirement

(III) Requirement not defined correctly

Errors → Leads to → Bugs → Leads to → Failure

Mistake — Defect / fault

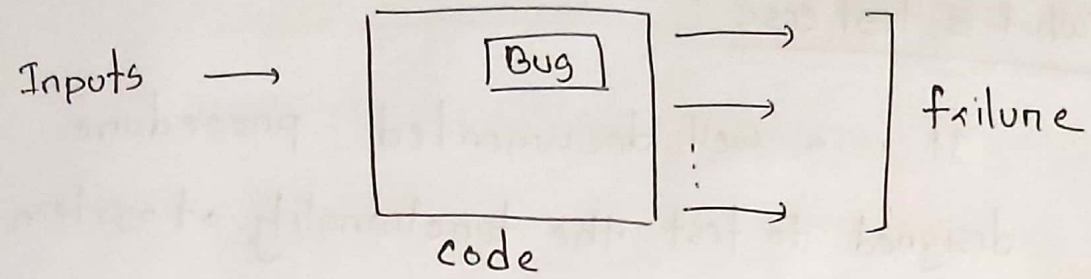
Def :- Error is a mistake in code that's made

why we cannot execute / compile code

Bug :- → is the term to describle the problems in a system output side

→ Bug is a condition that in actual causes a system to produce failure

→ unintended on unexpected behaviour in code



Failure :- → software doesn't perform expectedly

a failure is the inability of a software system or component to perform its required functions within specified performance Requirement

```
module A()
{
  ...
  while (a > n+1);
  {
    ...
    print(" The value : x");
  }
}
```

unexpected behaviour in code

Bug

Error
mistake by programmer

due to error condition is causing the loop to terminated prematurely and the code inside is not being executed

as it is not executing ; value x not printing

Failure

when software doesn't perform expected/ intended

What is test case ?

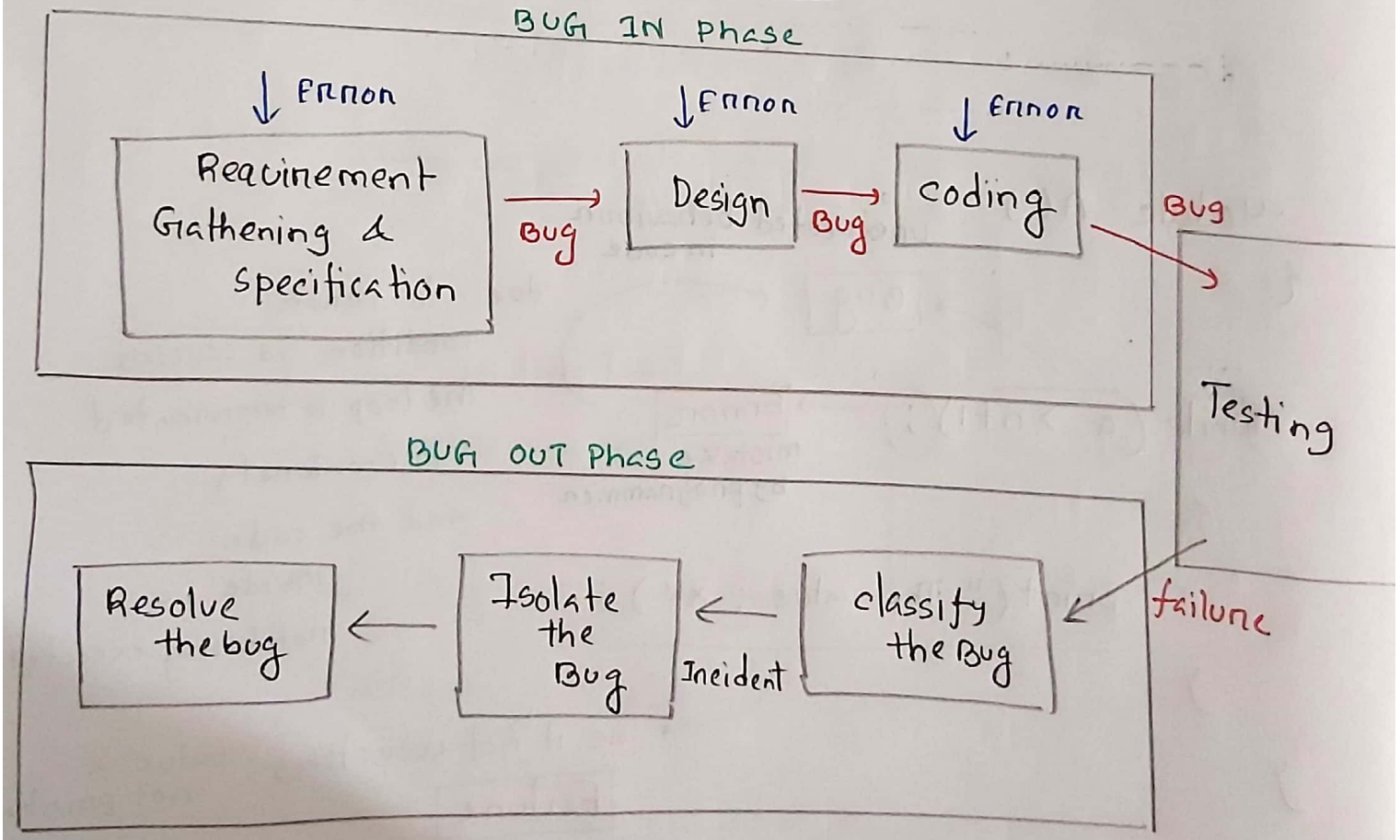
It is a well documented procedure designed to test the functionality of system.

- → Test case ID
- → Purpose
- → Pre conditions
- → Inputs
- → Expected outputs

] Properties of Testcase

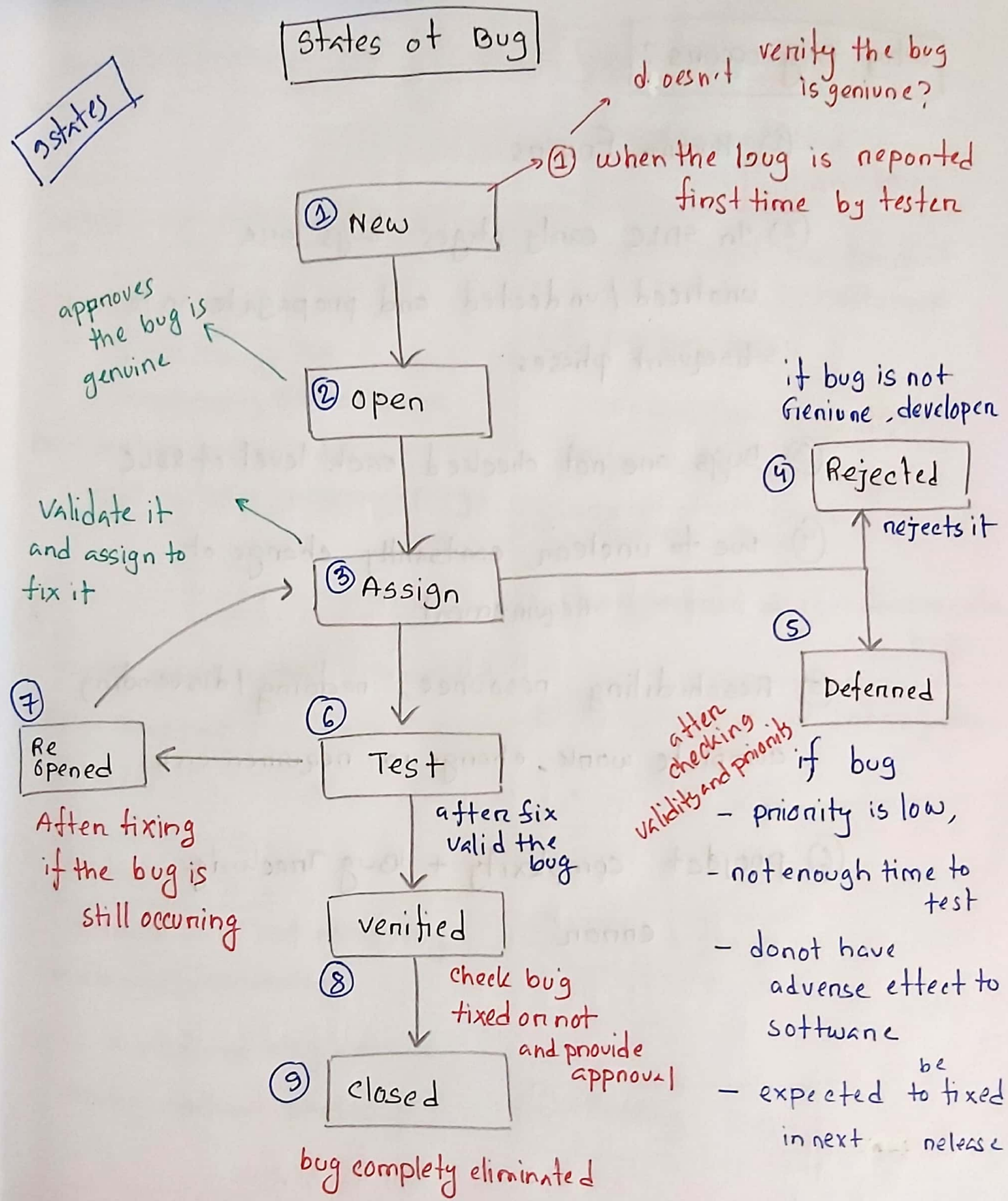
— o —

Life cycle of Bug



States of Bug

9 states



why Bug occors?

1. Human Ennons
2. In SDLC eanly stages bugs ane unoticed / undected and propagate into subsequent phases
3. Bugs ane not checked each level of SDLC
4. Due to unclean, contsantly change of Requinement
5. Reschudiling nesounee, nedoing / disconding complete work, change in nequinement
6. project complexity + Bug Tracking ennon

Bug classification

based on Criticality

1. **Critical Bugs**
   stops/hangs the functioning of software
   Ex:- sorting program where input is large program hangs

2. **Major Bugs**
   cause issue to fail requirement
   Ex:- Incorrect product price display

3. **Medium Bugs**
   outputs are not according to standard/conventions
   Ex:- A medium blog website showing irrelevant photos

4. **Minor Bugs**
   Typographical error, misaligned printout

based on SDLC

1. **Requirement and Specification Bugs**
   misunderstood the format needed by customer
2. **Design bugs**
   control flow bug, logic bugs, Data flow bug
3. **coding Bugs**
   typographical bugs, documenta- bugs
4. **Interface and Integration Bugs**
5. **system Bugs**
6. **Testing Bugs**

⊛ **Destructive approach** for testing

Bug hunting

→ Tester must acknowledge that bugs are always present in program and must think about the technique to solve it

→ This pshycology of being suspious about bug is a negative/destructive approach

→ help performing constructive and effective testing

→ the logic is to discover more and more bugs rather not to show system doesn't contain bugs. He should think like that there is always a bug in program and I need to find and solve it.

⊛ Benefits of early Testing

→ cost saving

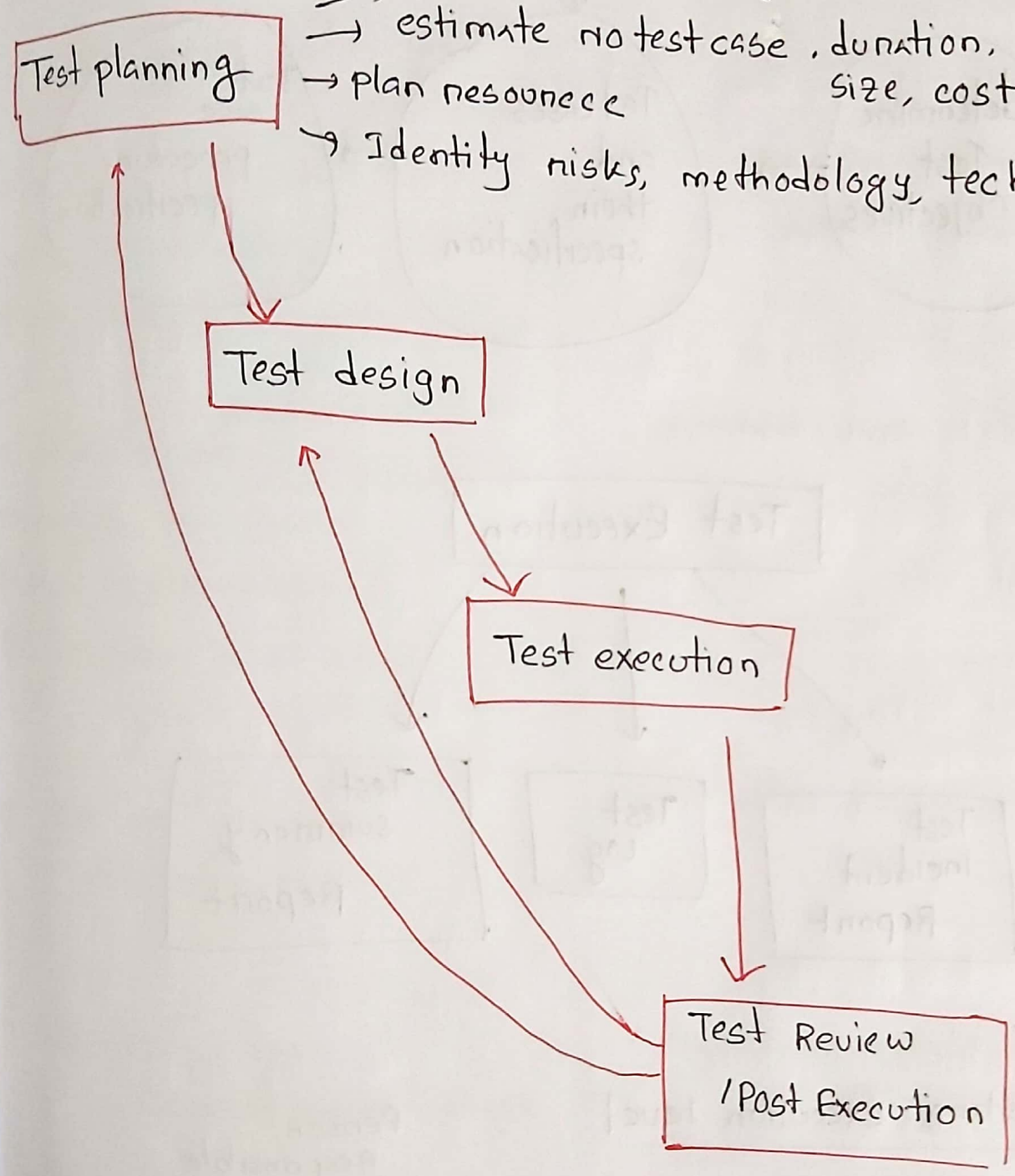
→ Defects Identification and correction

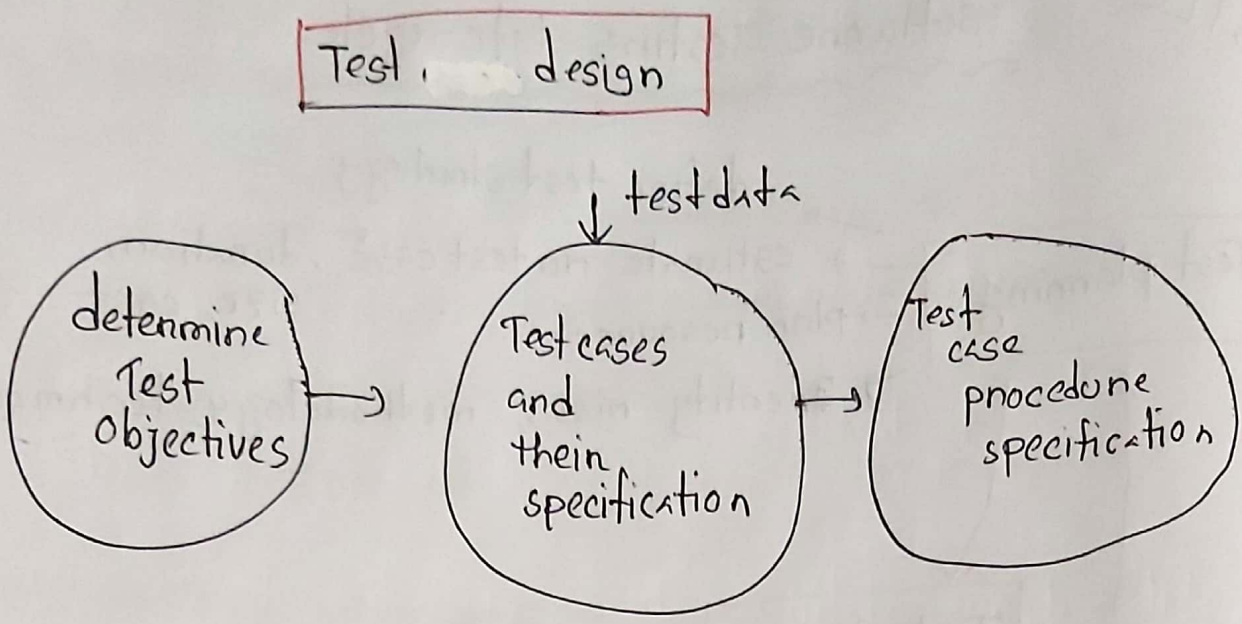
→ Improve Quality

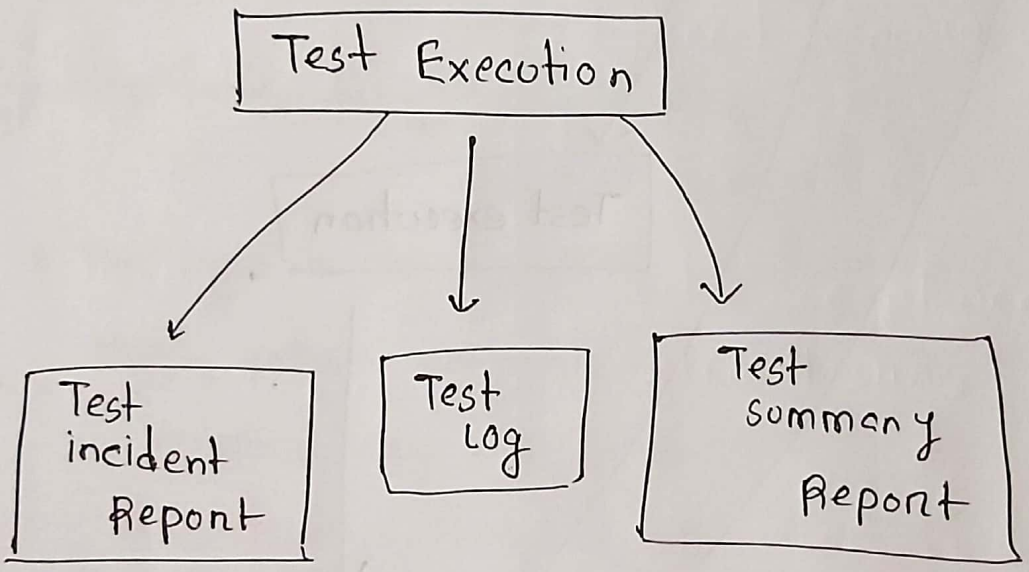
→ Fasten time to market

STLC

Software Testing Life cycle







Testing execution level	Person Responsible
Unit testing	→ developer
Integration testing	→ Tester + developer
System	→ " " + end user
Acceptance	→ Tester + End user

Testing strategy

(1) ensure product is designed and implemented connectly

Verification

→ to check software with its specification at every development phase

→ refers to set of activities that ensure connect implementation of functions in software

⇒ Are we building (2) the product right?

→ typically done (4) on development phase

→ goal :- To catch and fix (5) error at early stage

→ includes review, inspections, (3) walkthroughs

Validation

(1) → ensures that final product meets the specified Requirement and satisfy customer need

(3) → includes testing methods unit, system, integration

(2) → Are we building the right product?

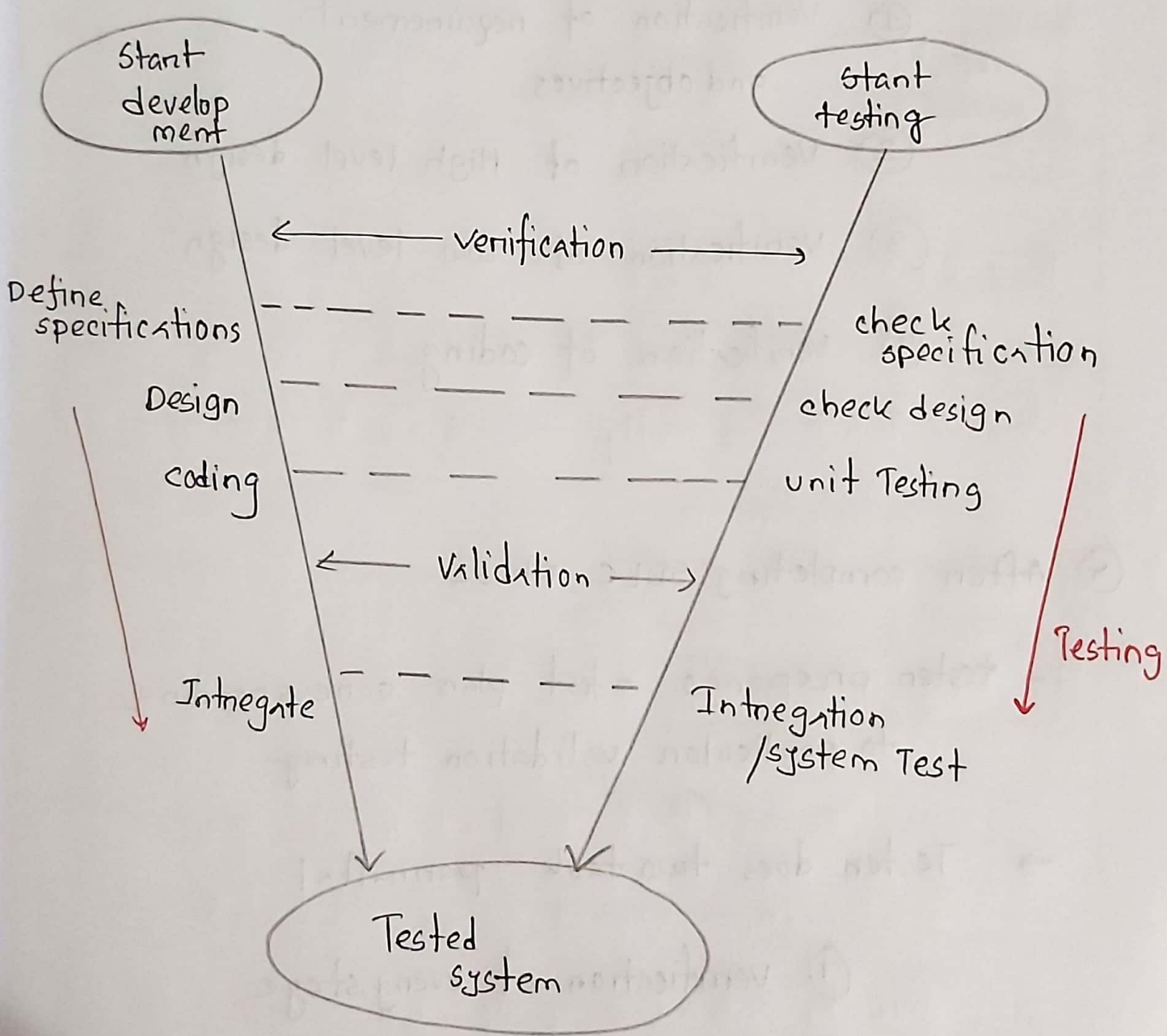
(4) → on testing phase

(5) → evaute system under dynamic condition and verify How it behaves in real life scenario

Static Testing	Dynamic Testing
a testing technique that review and analyze software documentation sourcecode and other antifacts before execution	that evaluates software behaviour when it is executed
objective :- To identify defects and improve software quality at early stage	to validate functionality and behaviour in different scenario
Types of testing - code review - walkthrough & Inspection	unit, acceptance system testing
Starts at early stage and continue in whole SDLC	starts after development

Chapter-3

Vtesting



(*) Verification is performed to under cover more and more bugs during earlier stages.

Verification activites are

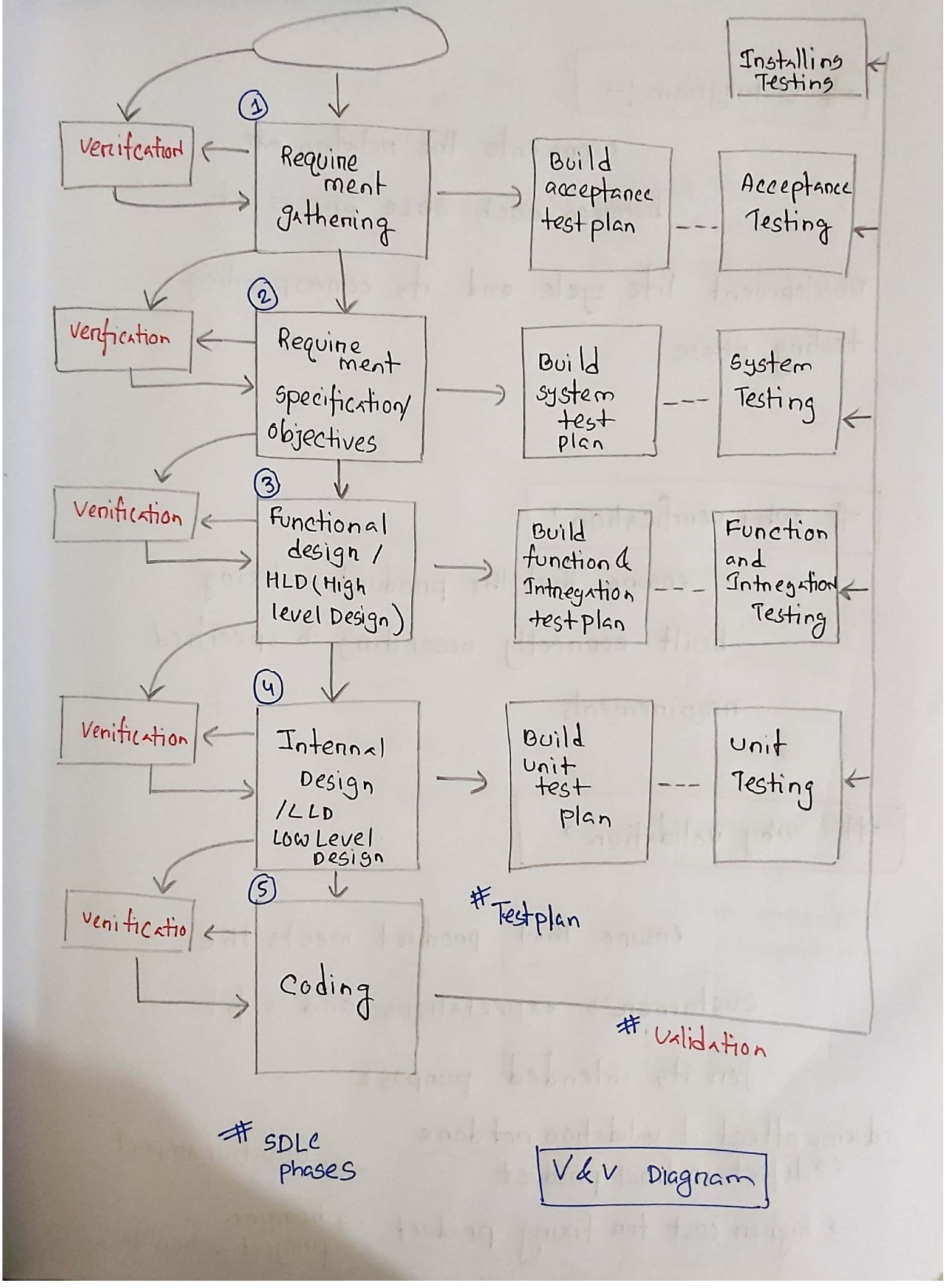
1. Verification of requirement and objectives
2. Verification of High level design
3. Verification of Low level design
4. Verification of coding

(*) After completing SDLC phase

→ tester prepares a test plan corresponding to particular validation testing

→ Tester does two task parallel

I. Verification of every stage
II. planning of validation


Installing Testing
1
Verification
Requirement gathering
Build acceptance test plan
Acceptance Testing
2
Verification
Requirement specification/ objectives
Build system test plan
System Testing
3
Verification
Functional design / HLD (High level Design)
Build function & Integration test plan
Function and Integration Testing
4
Verification
Internal Design /LLD Low Level Design
Build unit test plan
Unit Testing
5
Verificatio
Coding
Testplan
Validation
SDLC phases
V & V Diagram

V Diagram :-

represents the relation between each SDLC phases of Development life cycle and its conessponding testing phase.

why verification ?

ensure that the product is being built correctly according to specified requirements

why validation ?

ensure that product meets the customer's expectations and is fit for its intended purpose

adverse effect if validation not done

→ defect in final product → negative impact

→ higher cost for fixing product → hamper project schedule

Unit verification vs Unit Validation

checks whether the Unit meets its specification

ensures that the unit fulfills its intended purpose

Function testing → verifies Individual functions of software

vs

System Testing → verify and evaluates the entire system's compliance with specified requirement

vs

Acceptance Testing → ensure system meets user requirements