STQA

1) Evalution of Software testing

① Debugging oriented phase ② Demo oriented phase ③ Destruction oriented phase ④ Evaluation oriented phase ⑤ Pre-vention

⑥ Process oriented phase

x

Goals

Short term

Bug Discovery
Bug prevention

Reduced maintanence cost
Improved testing process.

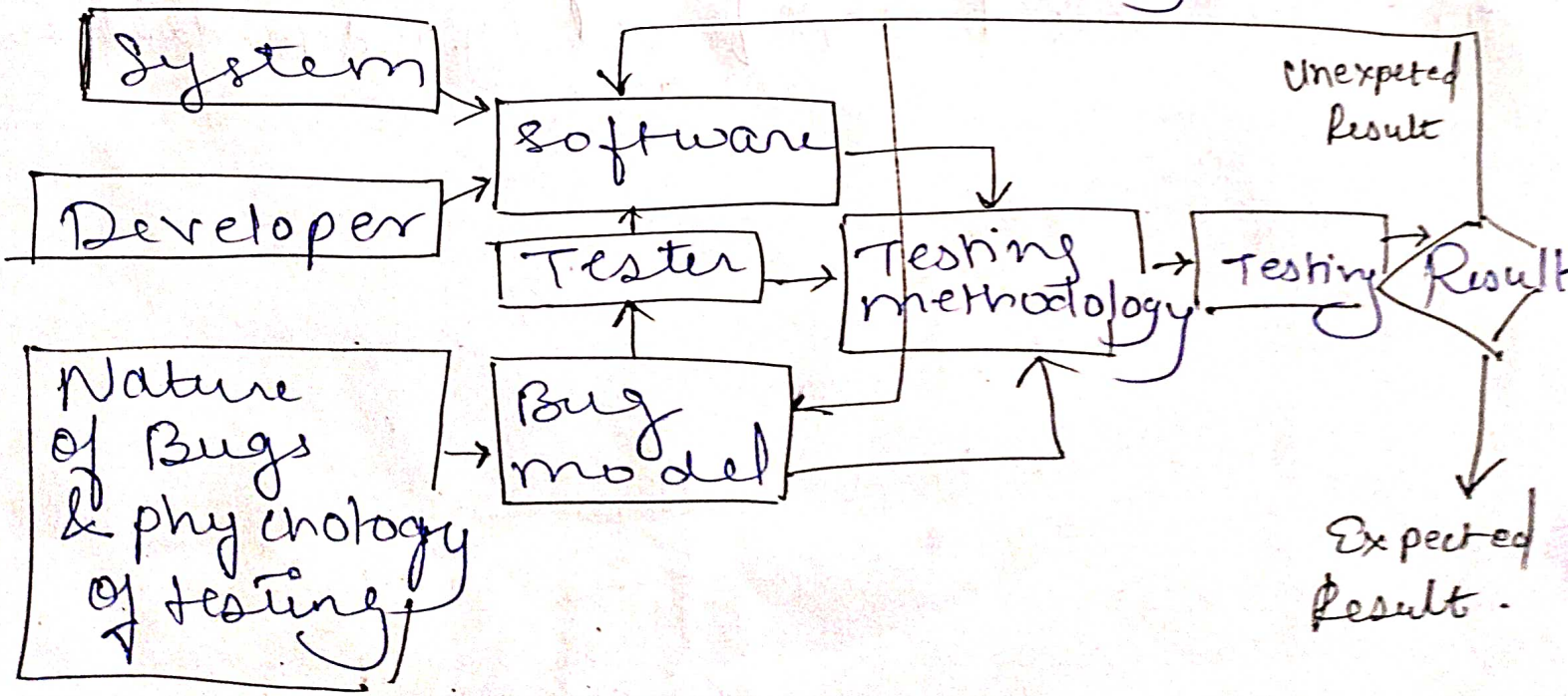
Software Testing

Reliability
Quality
Customer Satisfaction
Risk management

Post implementation

Long term

model for SW testing



Defination

1) failure - inability, not meeting specified expectations.
2) fault - condition that in actual causes failure
3) error - Dev team member makes mistak in any SDLC, errors are produced.

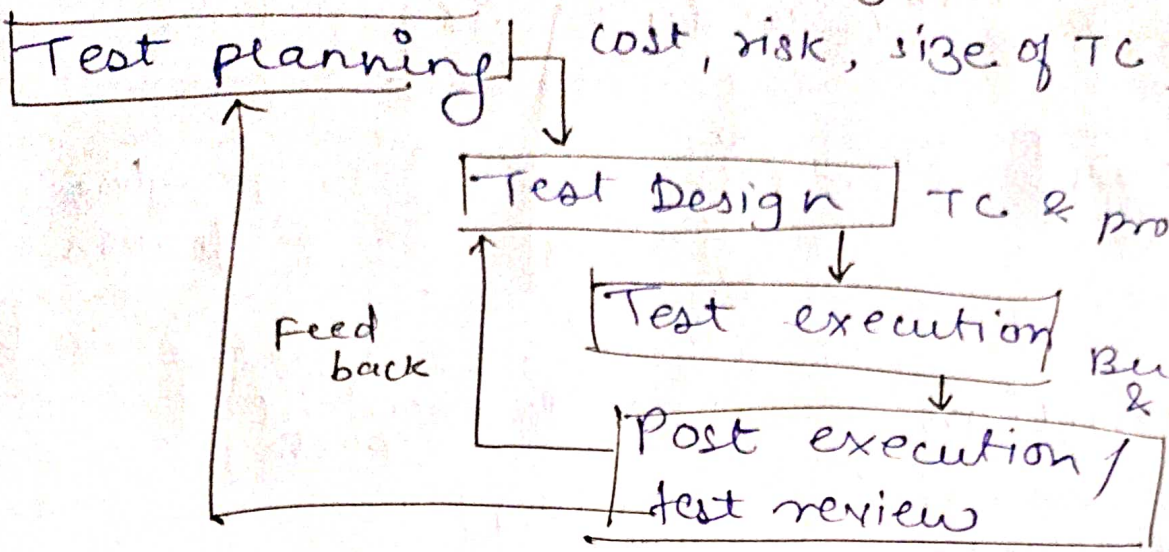
Error → Bug → Failures. Flow of faults

States of bugs (Not sure pg 47)

Testing Principles

1) Effective testing, not exhaustive testing
2) Testing is not a single phase performed in SDL
3) Destructive approach for constructive testing
4) Early testing is best
5) Existence of error & no. of error already found in section
6) Everything should be recorded
7) Testing should be performed by independant te

Software Testing Life Cycle (STLC).



① Test planning (7 points).

1) Defining test stragies.
2) Est tc, Duration & cost.
3) Identify areas of risk
4) Bug classification, bug severity, proj metrics.
5) Resource management
6) Methodologies & tools.

on analysis:-

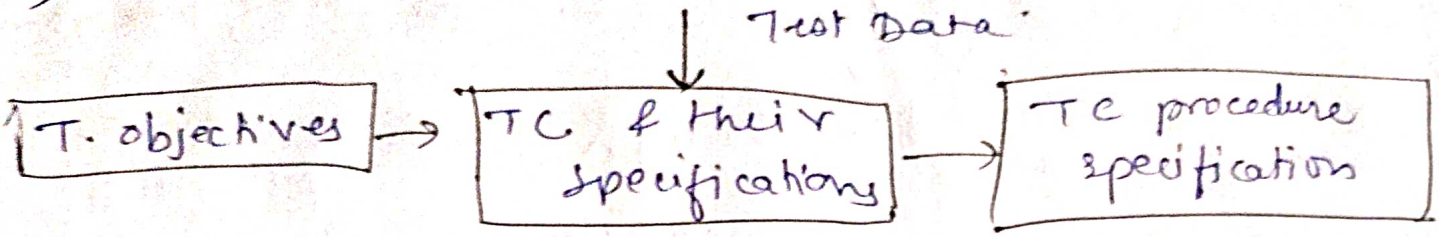
1) Develop TC format.
2) Identify TC to be automated
3) Prioritize TC
4) Define area of stress
5) Plan test cycles required for regression testing.
6) Develop TC plans acc to every phase of SDLC

Test Design.

1) Determining test objectives & prioritization
2) Preparing list of items to be tested
3) Mapping items to TC.
4) Selection of TC Design Techniques.
5) Creating TC & Test data.

6) setting up the test env & supporting tools.

7) Creating test procedure specification.



Test Execution

level	Responsible
Unit	Developer
Integration	Tester & Developer.
System	Tester, Developer, End users
Acceptance	Testers, End users.

T. incident report

T. log

T. summary report

Post Execution / T. Review.

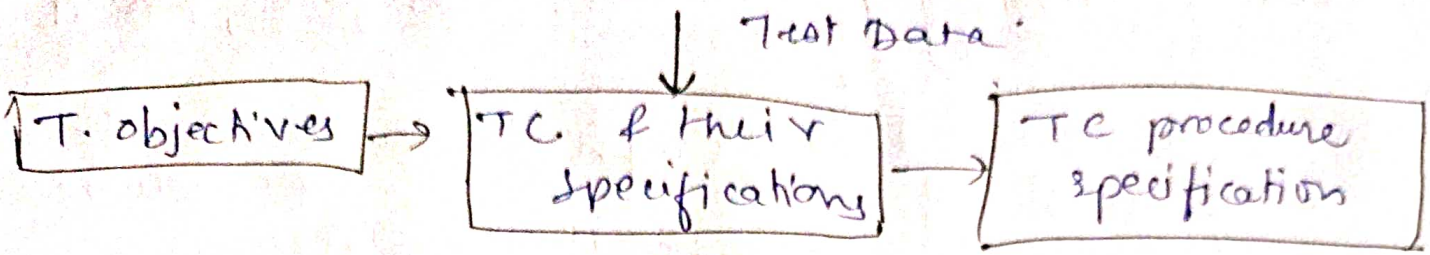
1) Understanding Bug
2) Reproducing Bug
3) Analyzing nature & cause of Bug.

after this & after analysing foll activities ca do

1) Reliability analysis.
2) Coverage analysis
3) Overall defect analysis.

6) Setting up the test env & supporting tools.

7) Creating test procedure & specification.



Test Execution

level	Responsible.
Unit	Developer
Integration	Tester & Developer.
System	Tester, Developer, End users
Acceptance	Testers, End users.

T. incident report

T. log

T. summary report

Post Execution / T. Review.

1) Understanding Bug 2) Reproducing Bug

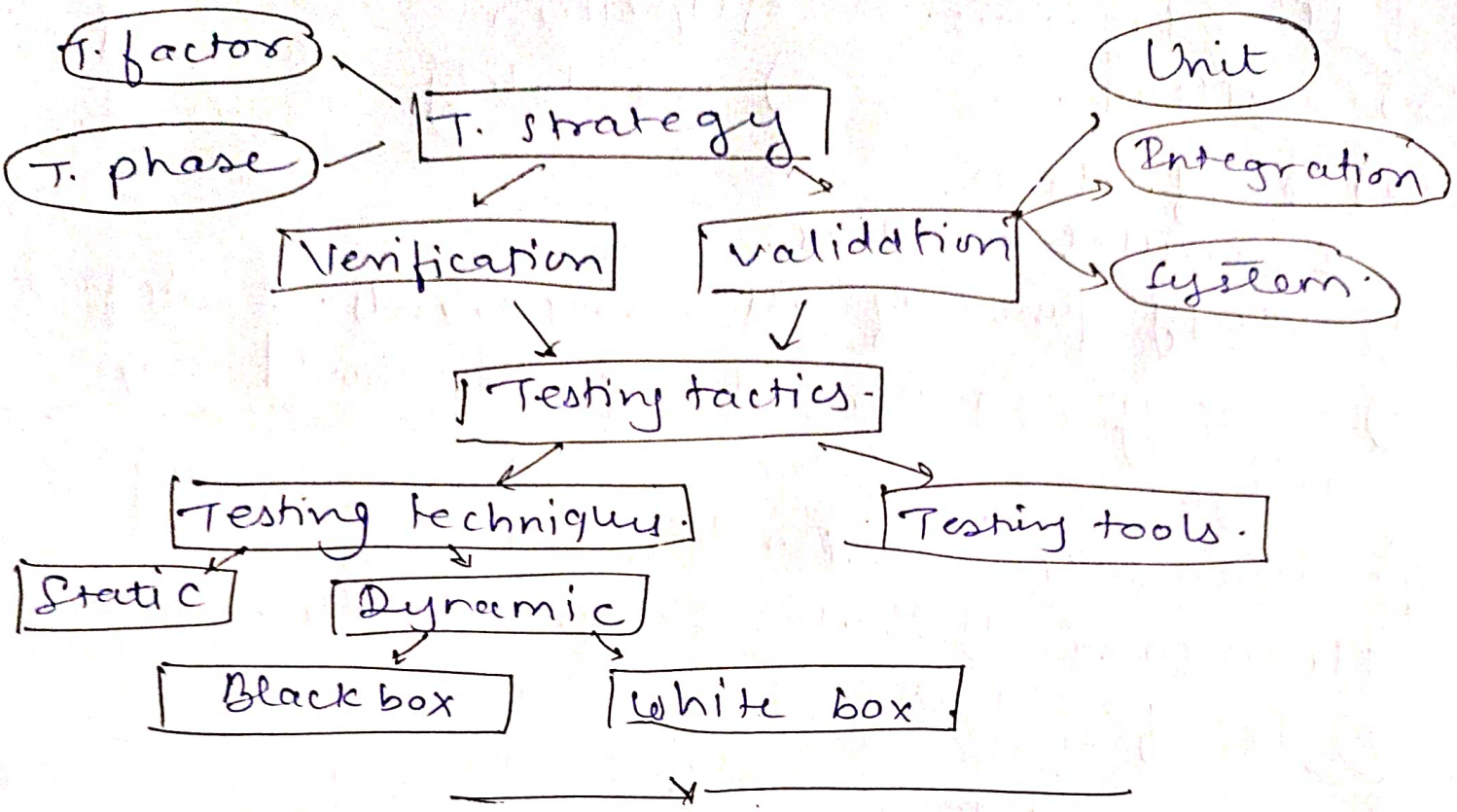
3) Analyzing nature & cause of Bug.

after this & after analyzing foll activities can be done:

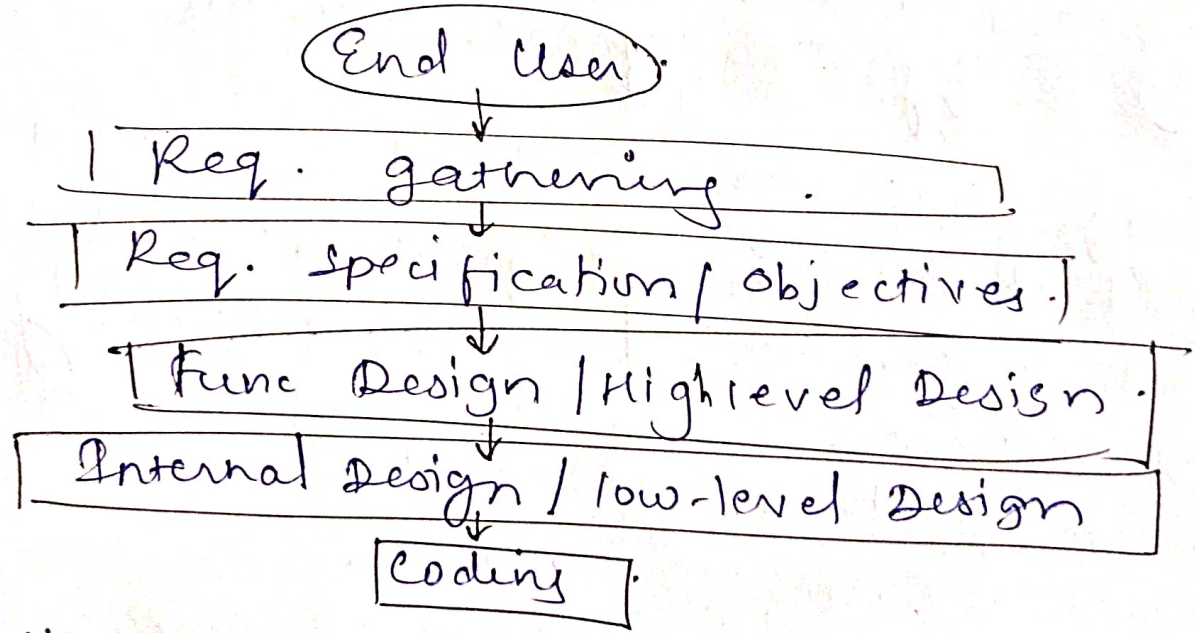
1) Reliability analysis. 3) Overall defect analysis.

2) Coverage analysis

Software Testing Methodology



Verification & Validation



HLD → 1) List of modules. 2) Brief functionality of each module
3) Overall arch. diagrams along with technology detail
4) Functionality with set of ext interfaces.
5) Interface relationships among modules inc. dependencies bet^n mods, DB tables with key elems

Verification

A) Requirements.

1) Correctness.
2) Unambiguous.
3) Consistent.
4) Completeness.
5) Updation
6) Traceability — Backward, Forward.

B) Vf of HLD.

2 parallel activities.

1) verify func of all components. & interface
2) Every requirement of SRS should map Design
3) Also prepares Function test plan based on SR

How to :- 1) Data Design. 2) Arch Design 3) Interface Design

C) Vf of LLD

2 parallel activities.

1) verify SRS of each module 2) & SDD.

How to:-

1) Check Design Specification in HLD & LLD has been coded using traceability matrix.
2) Examine code against language Specification checklist.

code verification - 1) Mixed mode operation 2) misunderstood or incorrect arithmetic precedence 3) Incorrect initialisation 4) Precision inaccuracy 5) Diff data type 6) Failure to exit. 7) ∞ loop.

A) Static technique

B) Dynamic testing technique.

UNIT Verification :-

White Box oriented, Vf of mods.

Validation

1) Acceptance T. plan -
prepared in requirement phase according to acceptance criteria.

2) System test plan -
verify obj specified in SRS.

3) Function T. plan.
prepared in HLD phase. every type of functionality can be tested.

4) Integration T. plan.
prepared to validate the integration of all mods such that all their interdependencies are checked.

5) Unit test plan.
Prepared in LLD. T plan for each mod. separately.

Validation T. Exec.

1) Unit V. testing -

a) ∵ smaller block, attention is focused

b) if whole sw is tested at once, Diff to find bug. ∴ Debugging Easy with U.T.

c) Diff. module, Diff developer.

2) Integration testing -
combining & testing multiple components or mods togethen. intention is to uncover the bugs present when unit testing mods are integrated.

3) Function Testing -
Obj is to measure the quality of functional (business) components of system.