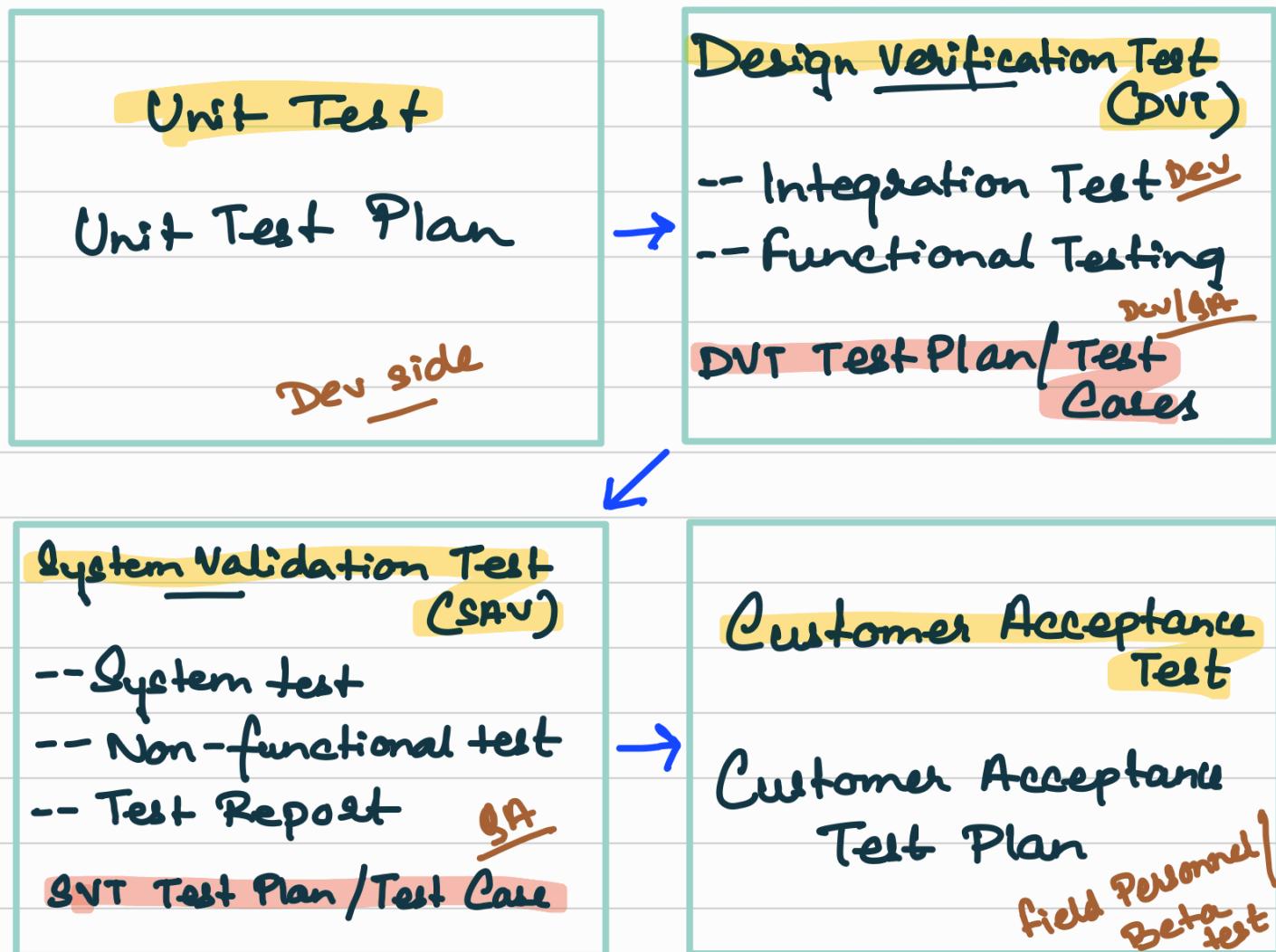


# L1 → Fundamental Testing Process

## Stages of Software Testing Process



## Test Plan Standard (IEEE)

IEEE Standard 829-2008.

A document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.

(Acc to Quality)

# Components of a Test Plan

The  
Assurance  
Institute

1. Testing approach / strategy
2. Scope
3. Schedule
4. Resources / Test environment
5. Entry & Exit Criteria.
6. Requirements Matrix (for Traceability)

## Traceability

- Tracing test cases to requirements.
- Tracing requirements to test cases -
  - i.e - if a req. changes which test cases are affected?
    - are all reqs. getting tested?
- Tracing either requirements or TCs to software modules.

## Test Plan Activities :

Use a Test Plan template or design one.

- List what **cannot** be tested.

- Write only what you need.
  - Have the Test Plan reviewed.
  - Make it a "living" document.  
*keep it updated*
- 

## Why we need a good test plan?

- organize, schedule & manage testing efforts
- Helps in writing test cases
- improves communication b/w dev. and management.

↑ Test Plan as a tool

- Measuring software quality is the intent (and must be planned)
- Developing good test sets take planning.
- Knowing when to stop
- More effective arguments when you have the facts.

↑  
As a product

## Concerns of Test Planning :

Not enough  
Training

Lack of  
customer/user  
requirement

Lack of  
management  
support

Not enough  
time

Lack of  
Test tools

Rapid Change

## Test Planning is IMPORTANT

1. Creating a plan & sticking to it makes it easier.
2. Ensure that everything is completed.
3. Know when to stop.
4. Have facts to bolster arguments.

5. Say **NO** to projects because if you have to.

---

## Test (Status) Report:

Summative vs. Cumulative info:

✓ Test Status Report → • How the test cycle is going?  
• Occurs after each testing cycle

✓ Test Report → • How the entire testing effort went for a project/feature  
• Occurs at end of testing.

→ What it should include?

- Evaluate how testing is going.
- List what was tested.
- List what was not tested and why.
- List still open defects
- Show the actual schedule.
- Tell developers what works/doesn't

→ Why is it required?

- Allows management & marketing to do their jobs.

- Assist in process improvement.

## In Short :

- List what was tested.
- List what wasn't tested.
- List still-open defects.
- List the schedule (actual & current)

\* Testing is independent but not isolated from development.

---

## Risk-based Test Planning :

Definitions associated with risk :

Risk: Any potential loss to an organization

Components of risk: Impact/Loss and Likelihood/Probability

Risk Analysis: Determining the impact and likelihood of various risks

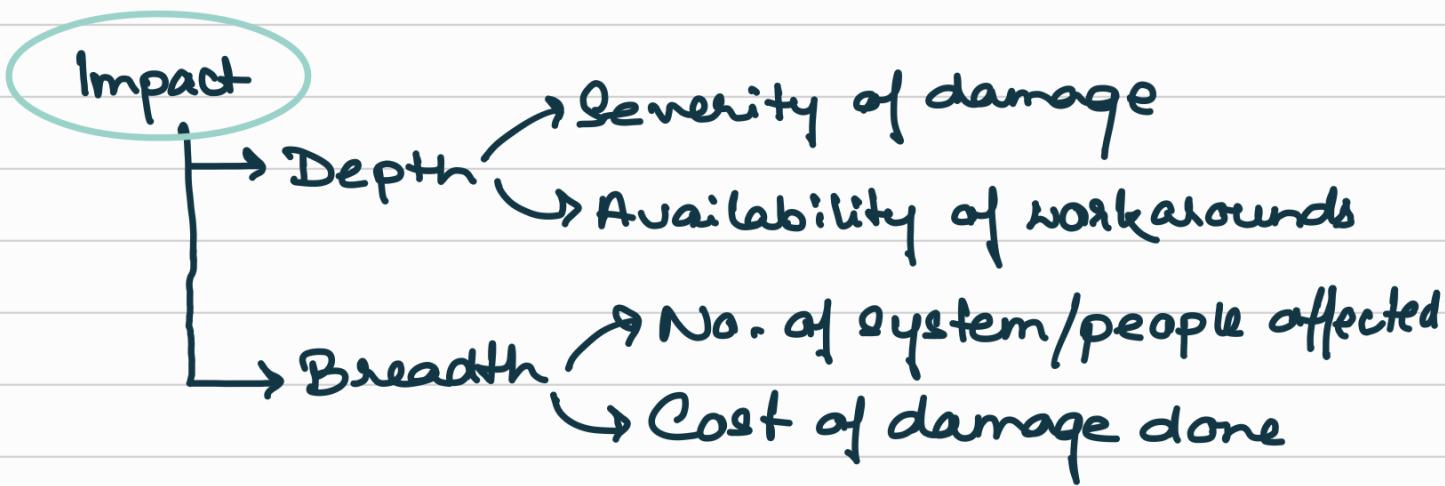
Risk Equation:  $(\text{Risk} = \text{Impact} \times \text{Likelihood})$

Risk Appetite: The amount of loss that management is willing to accept

Risk Mitigation: The act of reducing risk

## Nature of Risk :

2 components → • Impact  
• Likelihood



## Categories of impact:

- Financial
- Reputation
- Licenses
- Customer/user base
- Employees
- Lives

## Possible factors for likelihood of failures →

- Size of module
- Use of new(unproven) technology.
- Prior error history
- Complex features or modules
- Code that not has been unit tested

## Risk Mitigation: → act of reducing risk

- Risk avoidance
- Risk transfer (insurance, indemnification, etc)
- Risk Management
  - ↳ Loss prevention

↳ Crisis management  
↳ Training

## Conventional Approach to Risk-based Testing :

When time is running out, risk can help prioritize

IMPACT → Test those features with the highest impact

LIKELIHOOD → Test those features most likely to fail

Design your tests on 2 factors —

1. What can go wrong?
2. What must go right?

.

---

