Unit 1: Introduction

Lucky Page No.: Date: / /

* Some Software failures.

- A major problem of the software industry is its inability to develop error-free software.
- S/w crises has become a fixture of everyday life with many well-publicized failures that have had not only major economic impact but also have become the cause of loss of life.
- Some of the failures are listed below

i) The explosion of the ariane 5 rocket.
ii) The Y2K problem
iii) The USA star-wars program
iv) Failure of london ambulance system
v) USS yorktown incident
vi) Accounting software failures.
vii) Experience of windows XP

i) The explosion of the ariane 5 rocket
- This loss of information was due to specification & design errors in the software of the inertial reference system.
- A s/w fault in the inertial reference system was identified as a reason for the explosion by the enquiry committee.

Lucky Page No.:
Date : / /

ii) The Y2K problem.

- The Y2K problem was the most critical problem of the last century.
- It was simply the case of using 2 digits for the year instead of 4 digits.
- For eg. 1965 was considered as 65.
- The developers could not imagine the problem of year 2000: what would happen on Jan 1, 2000? The last 2 digits i.e 00 may belong to any century like 1800, 1900, 2000 etc.
- The simple ignorance / a faulty design decision to use only the last two digits for the year resulted into the serious Y2K problem.

iii) The USA Star Wars Program

- 'patriot missile' was the result of the USA 'Star Wars' program.
- This missile was used for the first time in the Gulf war against the Scud missile of Iraq.
- Surprisingly, 'patriot missiles' failed many times to hit the largest Scud missile.

iv) Failure of London Ambulance System

- The software controlling the ambulance dispatch system of London collapsed on October 26-27 1992 & also on Nov. 4, 1992 due to s/w failures.
- The London Ambulance Service was a challenging task that used to cover an area of 600 square miles & handled 1500 Emergency calls per day.

18, 1900, 2000, 2100

v) USS yorktown Incident
- The USS yorktown a guided missile cruiser was in the water for several hours due to the software failure in 1998.
- A user wrongly gave 0 value as an i/p which caused a division by zero error
This fault further failed the propulsion system of the ship & it did not move in the water for many hours
- The reason behind this failure was that the program did not check for any valid i/p.

vi) Accounting slw failures.
- Financial slw is an essential part of any company's IT infrastructure.
- However, many companies have suffered failures in the accounting system due to errors in financial slw.
- The failures range from producing the wrong info to the complete system failure, there is widespread dissatisfaction over the quality of financial slw.

vii) Experience of windows XP
- charles c Mann shared his views about windows XP through his article in technology reviews:
"Microsoft released windows XP on October 25, 2001.
That same day, what may be a record, the company posted 18 megabyte of patches on its website for bug fixes, compatibility updates & enhanced.

Lucky Page No.:
Date : / /

* Testing process

- Testing is an important aspect of the s/w development life cycle.
- It is basically the process of testing the newly developed s/w, prior to its actual use.
- Testing is very expensive process & consumes one third to one half of the cost of a typical development project.
- It is largely a systematic process but partly intuitive too.

i) What is Software testing

Good testing entails more than just executing a pgm with desired input(s). Let's consider a pgm named as 'minimum' (shown in below pgm) that reads a set of integers & prints the smallest integer.

```
void minimum();
void main()
{
    minimum();
}
void minimum()
{
    int array[100];
    int number;
    int i;
    int temp data;
    int Minimum = int MAX;
    clrscr();
    PF("Enter the size of the array");
```

```
Sf("%d", &Number);
for (i=0; i<Number; i++) {
Pf("Enter A[%d]=", i+1);
Sf("%d", &tmpdata);
tmpdata = (tmpdata<0) ? -tmpdata : tmpdata;
array[i] = tmpdata;
}
i=1
while (i<Number-1) {
    if (minimum > array[i])
    {   minimum = array[i];
    }
    i++;
}
Pf("Minimum = %d\n", Minimum);
getch();
}
```

Test case	size	i/p's seq integers	Expected o/p	observed o/p	Match
1	5	6, 9, 2, 16, 19	2	2	yes
2	7	46, 11, 32, 9, 39, 99, 91	9	9	-11-
3	7	89, 36, 42, 16, 65, 76, 81	16	16	-11-
4	6	28, 21, 36, 31, 30, 36	21	21	-11-
5	6	106, 109, 88, 111, 114, 116	88	88	-11-
6	6	61, 89, 99, 31, 21, 69	21	21	-11-
7	4	6, 2, 9, 5	2	2	-11-
8	4	99, 21, 7, 49	7	7	-11-

table: i/p & o/p of the pgm minimum

There are 8 sets of I/P's, in all these test cases the observed O/P is the same as expected O/P.

∴ There r many def^n of testing A few of them are-

1. Testing is the process of demonstrating that errors are not present.
2. purpose of testing is to show that a pgm performs its intended fun^n correctly
3. It is process of establishing confidence that a pgm does what it is supposed to do.

ii) Why should we Test ?

- Slw testing is a very expensive & critical activity, but releasing the S/w without testing is definitely more expensive & dangerous.
- It is like running a car without brakes.

∴ testing is essential ; but how much testing is required

- Do we have methods to measure it ?
- Do we have techniques to quantify it ?

The Answer is not Easy. All projects are diff^n in nature & functionalities & a single yardstick may not be helpful in all situations.

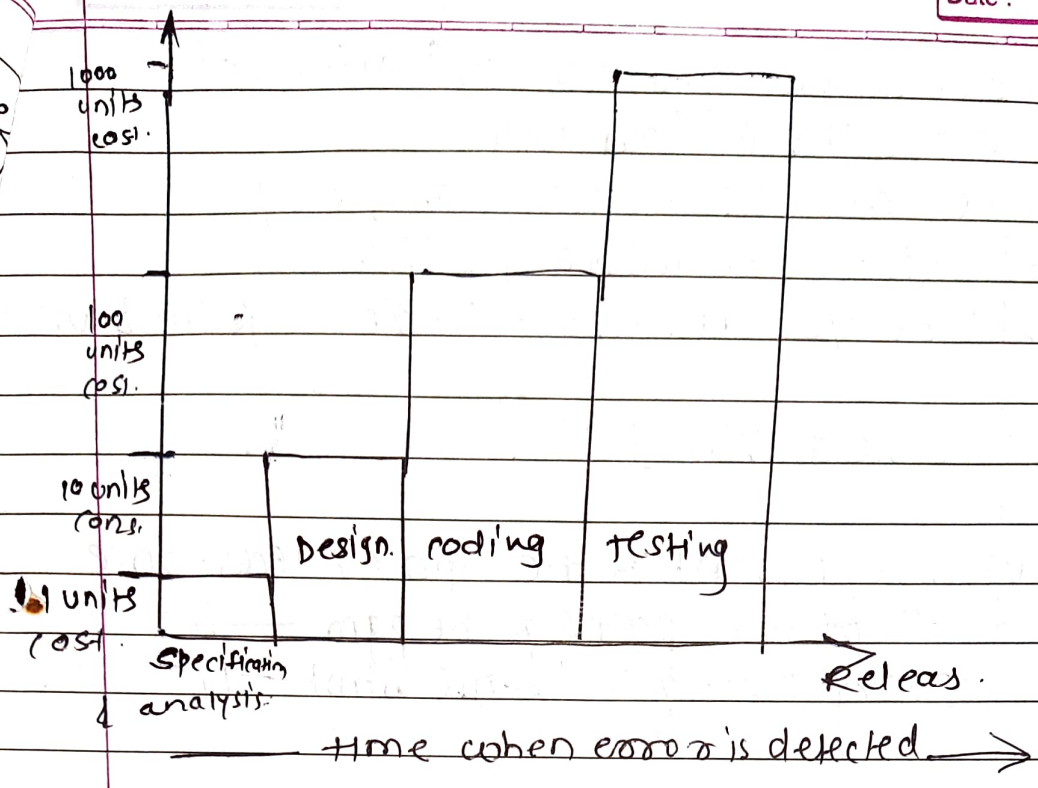


Fig. Phase wise cost of fixing an error.

iii) Who should we do the testing ?

- Testing a s/w system may not be the responsibility of a single person. Actually, it is a team work. The size of the team is dependent on the complexity, criticality & functionality of the s/w under test.
- The s/w developers should have a reduced role in testing, if possible.
- The testing persons must be cautious, curious, critical but non judgmental & good communicators.
- One part of these job is to ask questions that that the developers might not think.

Some questions are

1. How is the s/w ?
2. How good is it ?

3. How do u know that it works? Evidence?
4. What are the critical area?
5. What are the weak areas & why?
6. —— 11 —— serious design issues?
7. what do you feel about complexity of the source code?

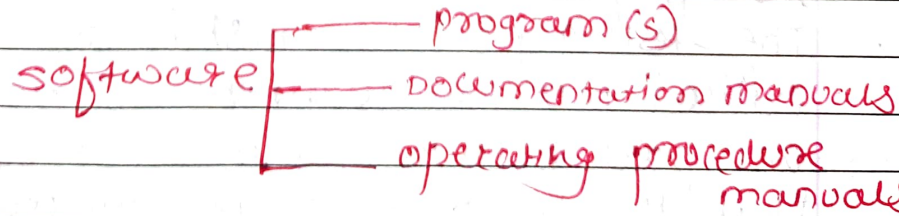
iv) What should we test?

- Execute every Stmt of the pgm at least once.
- —11— all possible paths of the pgm —11—
- —11— Every exit of the branch Stmt —11—

* Some Terminologies.

i) program and Software.

- Both terms are used interchangeably, although they are quite different.
- The SW is the superset of the program(s).
- It consist of one or many program(s), documentation manuals & operating procedure manuals.



software = pgm (s) + Docu. manual + operating proc. manual.

Fig: Components of the Software.

- The pgm ~~is~~ Source code and object code.

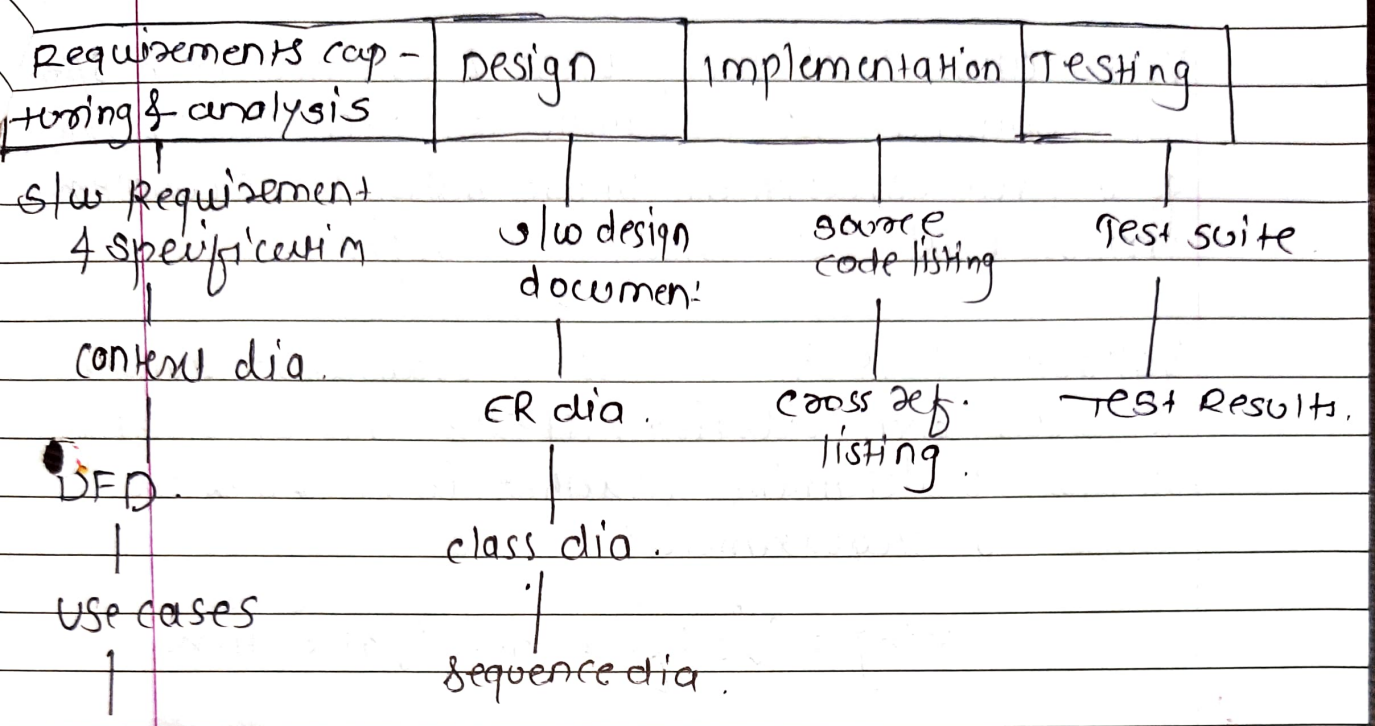


Fig :- Documentation Manuals.

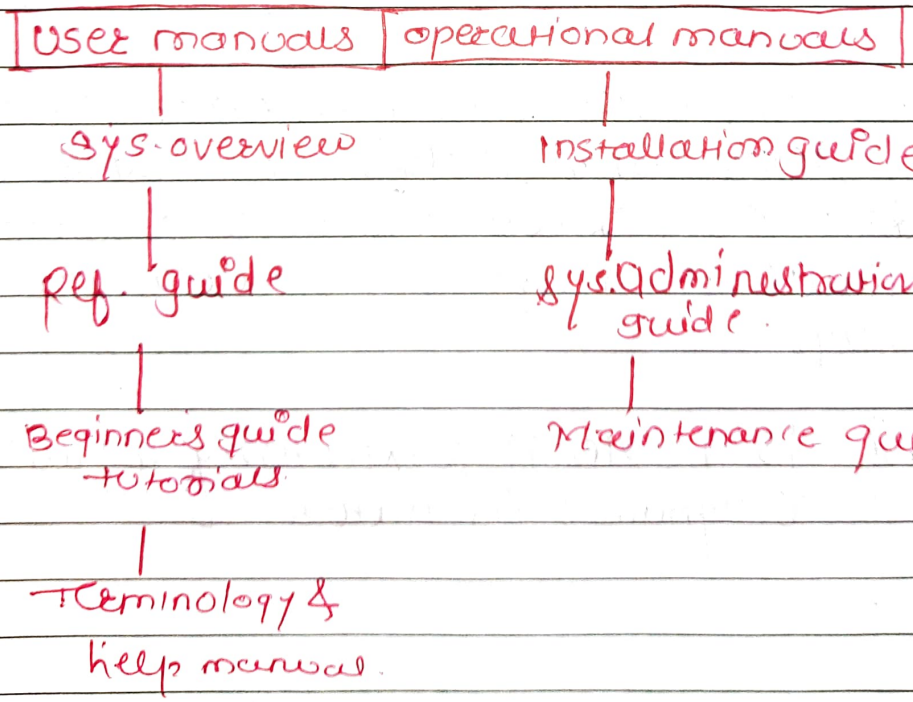


fig. OS manuals.

Is product being built the right way? e.g. Order chicken biryani ↓ Valid^n
verify
- Is it look like biryani? How does it smells? Are there chicken pieces?
How actually it tastes?

Lucky | Page No.: | Date: / /

ii) Verification & Validation

Verification: Is the process of Evaluating the System or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

Validation: Is the process of Evaluating a System or component during / at the end of development process to determine whether it satisfies the specified requirements.

Testing = Verification + Validation.

iii) Fault, Error, Bug & failure

Error - When we make an error during coding we call this a 'bug'.

Fault - Is representation of an error where representation is the mode of expression such as ~~data~~ DFD, ER dia, source code, use cases. If fault is in the source code, we call it bug.

Failure: Is the result of execution of a fault & is dynamic in nature.

Lucky | Page No.:
Date : / /

iv) Test, Test case and Test suite

Test case Identification Number:

Part I (Before Execution)

1. purpose of test case:
   pre condition(s):
2. (optional).
3. input (s):
4. Expected output (s):
5. post condition(s):
6. Written by:
7. Date of design.

Part II (After Execution)

1. output (s):
2. post condition(s):
   (optional)
3. Pass / Fail:
4. if fails, any possible reasons of failure (optional)
5. Suggestions (optional)
6. Run by:
7. Date of suggestion.

v) Deliverables and Milestones.

- Diff. deliverables are generated during various phases of the SDLC development eg. Source code, SRS, SDD, Installation guide, User ref. manual, etc.
- Milestones are the used to ascertain the status of the projects.

Lucky | Page No.:
Date : / /

vi) Alpha, Beta and Acceptance Testing

Acceptance Testing : This term is used when the s/w is developed for a specific customer.

Alpha & Beta Testing : These terms are used when the s/w is developed as a product for anonymous customers.

vii) Quality & Reliability

- s/w reliability is one of the imp factors of s/w quality. Other factors are understandability, completeness, portability, consistency, maintainability, usability, efficiency etc. These quality factors are known as non functional requirements for a software system.
- s/w reliability is defined as "the probability of failure free operation for a specified time in a specified env."

viii) Testing, Quality Assurance & Quality Control

- The purpose of QA activity is to enforce standards & techniques to improve the development process & prevent the previous faults from ever occurring.
- Quality control attempts to build a s/w system & test it thoroughly.

Lucky Page No.:
Date : / /

IX Static and dynamic Testing.

Static Testing : Refers to testing activities without executing the source code.

Dynamic Testing : Refers to executing the source code & seeing how it performs with specific

X Testing & Debugging

- The purpose of testing is to find faults & find them as early as possible
- When we find any such fault the process used to determine the cause of this fault & to remove it is known as debugging.

* Limitations of Testing

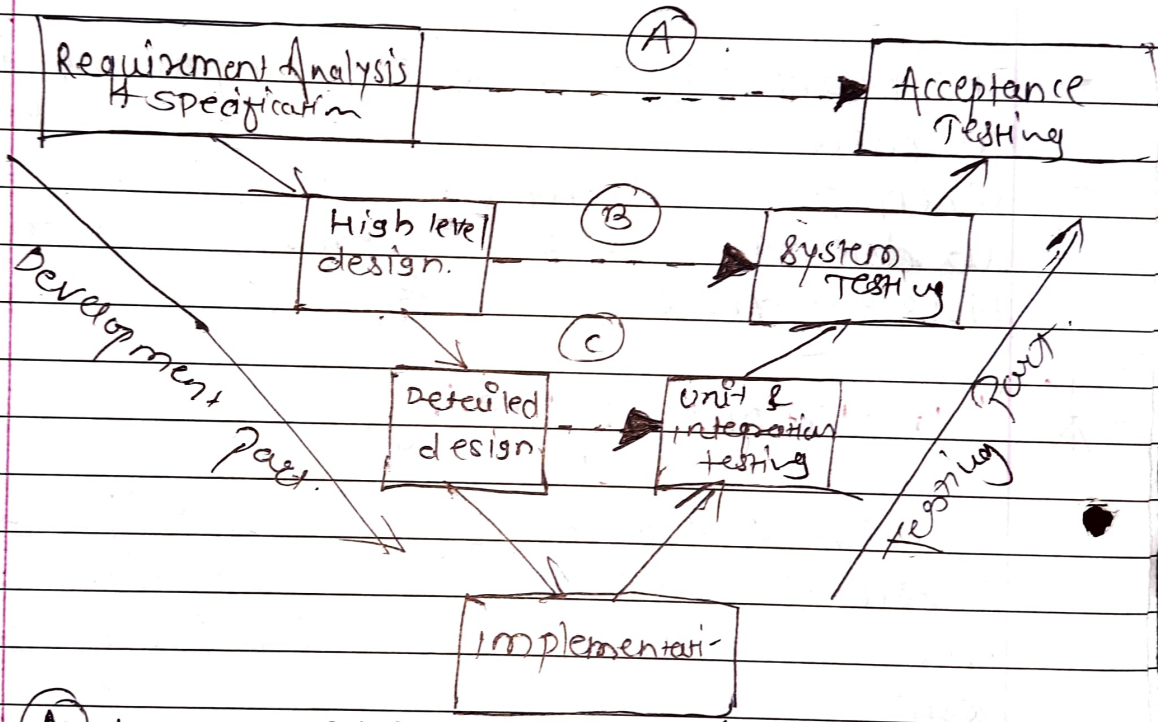
- Test everything before giving the slw to the customers. When we say 'everything', we may expect one, two / all of the following

i) Execute every statement of the program
ii) Execute every true & false condition
iii) Execute every condition of a decision node
iv) Execute every possible path
v) Execute the program with all valid inputs
vi) Execute the program with all invalid inputs

i) Errors in the S/w requirement & specification documents.
ii) Logical Bugs
iii) Difficult to measure the progress of Testing.

* The V Shaped Software life cycle model.

i) Graphical Representation



(A) Acceptance test case design & planning
(B) System test case design & planning
(C) Unit and integration test case design & planning

Figure: V Shaped S/w development life cycle model.

Lucky | Page No.:
Date : / /

Relationship of development & testing parts.

- The development part consists of first 4 phases, i.e.
  i) Requirement analysis & specification
  ii) High level design
  iii) Detailed design
  iv) Implementation

- The testing part has 3 phases
  i) Unit & Integration testing
  ii) System tesing
  iii) Receptance Tesing.