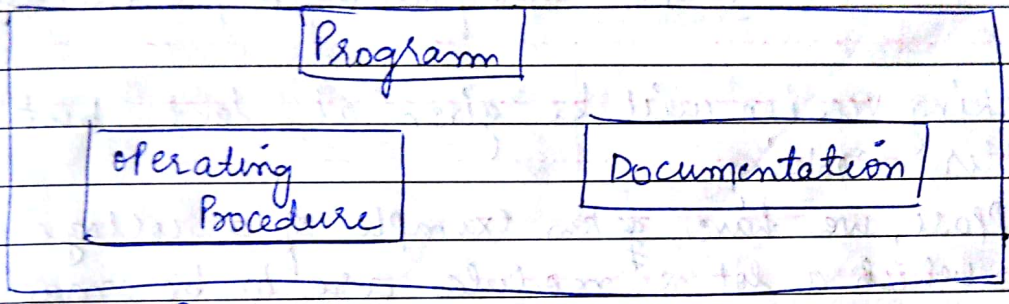
Date __/__/____ Page No.: ____

Software Engineering

A discipline whose aim is the production of quality software, software that is delivered on time within budget and that satisfies its requirements.

* SOFTWARE is a collection of Programs.



SDPM = Software Development Process Models.

1) Waterfall Model: → Sup (small)

Requirement analysis & Specification → Design → Implementation & Unit Testing → Integration & System Testing → Operation & Mentainence

SRS

* SYSTEM is a combination of Software and hardware.

"There are no great things, only small things with great love. Happy are those." - Mother Teresa

3. For large projects this model is not suitable as large projects require a lot of time to complete, so the [illegible] user [illegible] used early may not be feasible during the [illegible] time and it may become obsolete (obsolete).

4. Since the user requirements change day by day [illegible] [illegible] not sure about the requirements.

★ Disadvantages of Waterfall Model :

1. No roll back.
If we have started from the initial block and reached till the middle, then if we want to go back to the first position from b/w the procedures, this is NOT POSSIBLE i.e. IT IS SEQUENTIAL. (follows a particular sequence)

2. Working version will be given at last but only after working.
Suppose, we have taken example of a college system in which a lot of modules are to be made, then if I want that if any one of the module is finished so it should be given to ~~me~~ the user so that ~~I~~ he can start working on it, but this is NOT POSSIBLE bcoz once the whole model is complete only then it will be given, because if the individual model is working perfectly then it is not important that when all the modules will be put together then also it will work.

2) Increment Process Model :→ These models are the models which are given on a very short notice to the user. Here we give continuously give a limited functionality system of the required system to the user if he/She has to get it on a very short notice, every time we give the model with more efficient functionalities.

A) Iterative Enhancement Model.

But final ke liye vo customer ke subset hi fulfill kar raha ya (PTO)

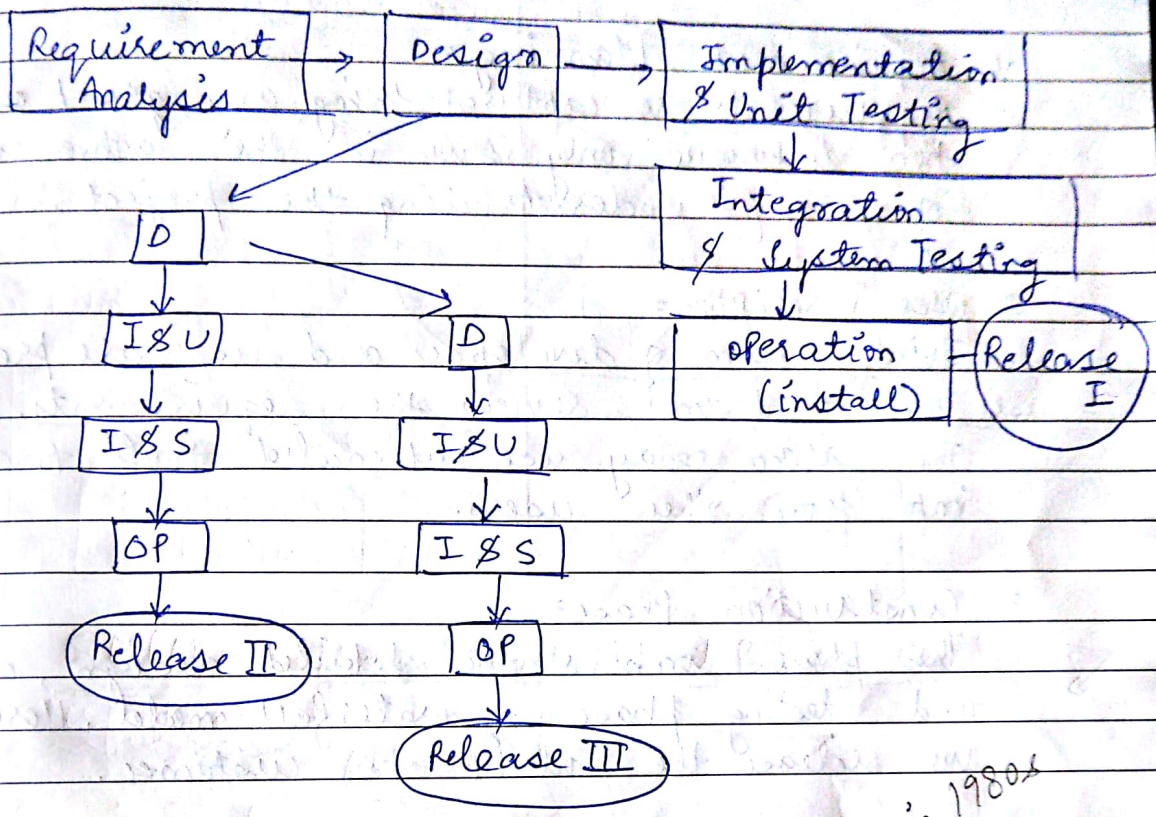
"Patience is a key element of success." – Bill Gates

Date __/__/____

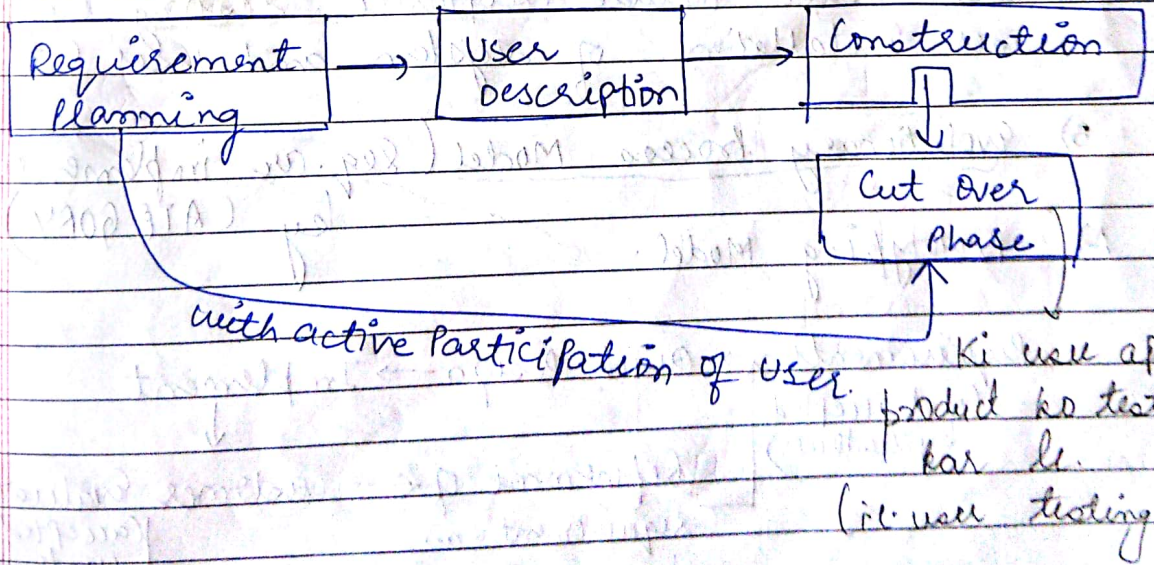
functional requirement
issue book
return book

non-functional requirement
maintainability
flexibility

Page No.: _____



Rapid Application Development (RAD) : IBM in 1980s



Ki user apne product ko test kar le.
(i.e. user testing).

"There are no great things, only small things with great love. Happy are those." - Mother Teresa

Date __/__/____ Page No.: ____

1. Requirement Planning:
Requirements are captured using any group elicitation technique, only issue is the active involvement of users for understanding the project.

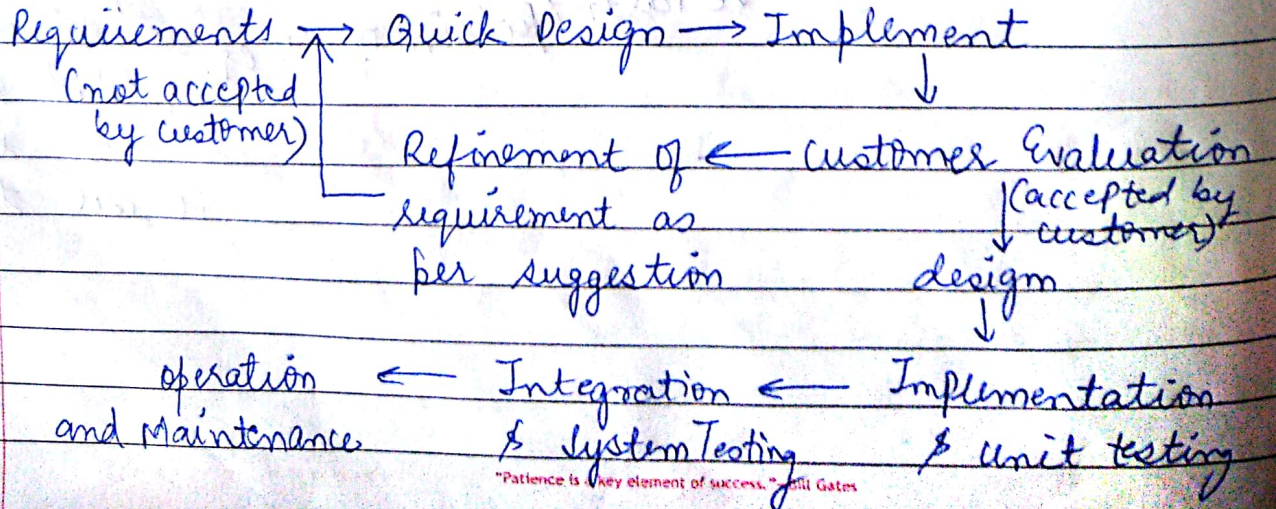
2. User Description:
Joint team of developers and user are prepared, user understand and review the requirements.
The team may use automated tools to capture info from the users.

3. Construction Phase:
This phase combine the detailed design, coding and testing phase of waterfall model. Here we release the product to customer.

4. Cut Over Phase:
This phase include exceptance testing by the user Installation of system and user training.

B) Evolutionary Process Model (Req. are implemented by CATEGORY).

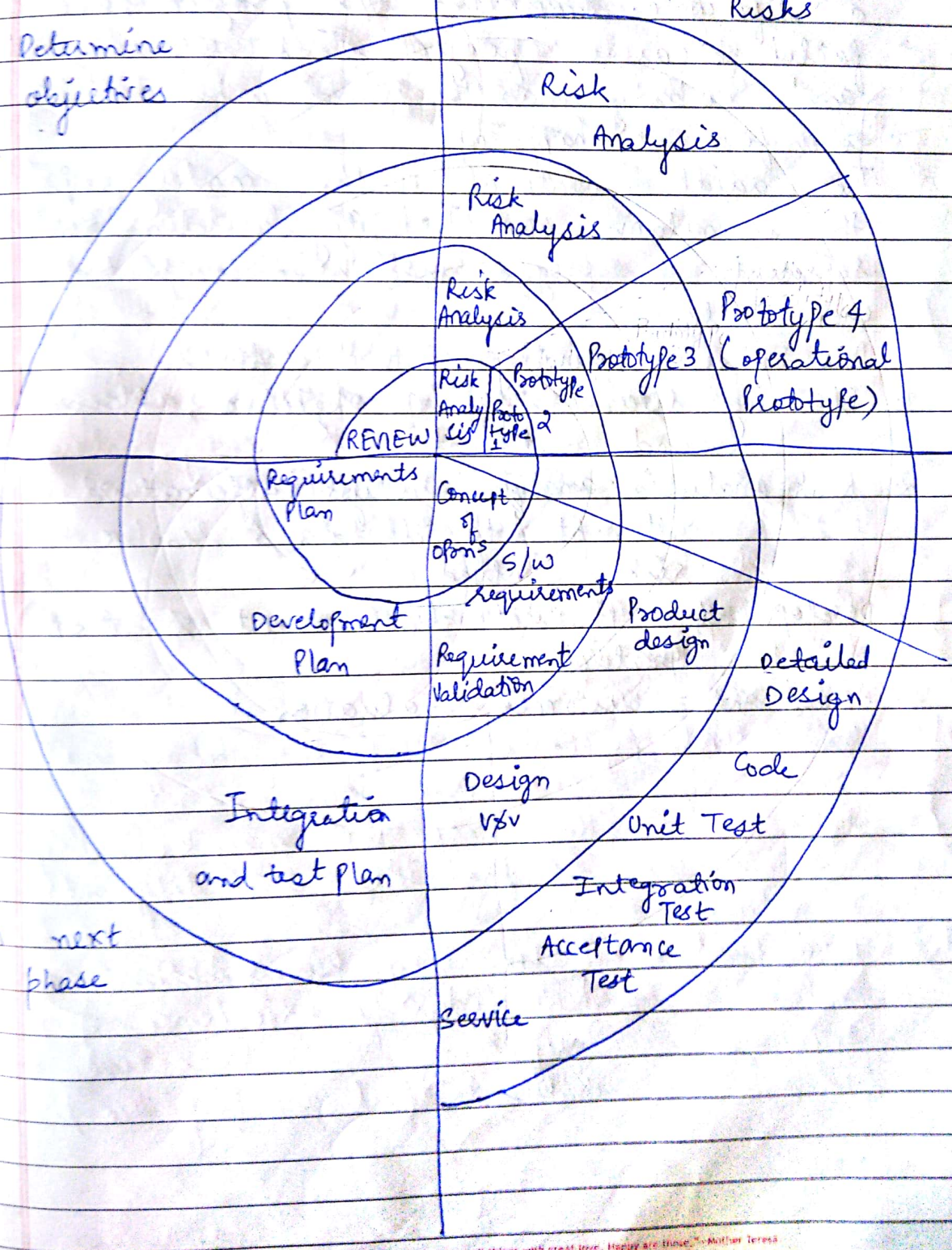
A) Prototyping Model.



"Patience is key element of success." Bill Gates

Date ___/___/_____ Page No.: _____

SPIRAL MODEL



"There are no great things, only small things with great love. Happy are those." - Mother Teresa

Date __/__/____ Page No.: HPSC

→ Imp.

Spiral Model: (RISK)

The problem with traditional s/w model is that they do not include project risk factor into a life cycle. Important s/w project have failed because project risk were neglected and nobody was prepared when something unforeseen happened.

The radial dimentional of the model represent the cumulative cost. The angular dimention represent the progress made in completing each cycle.

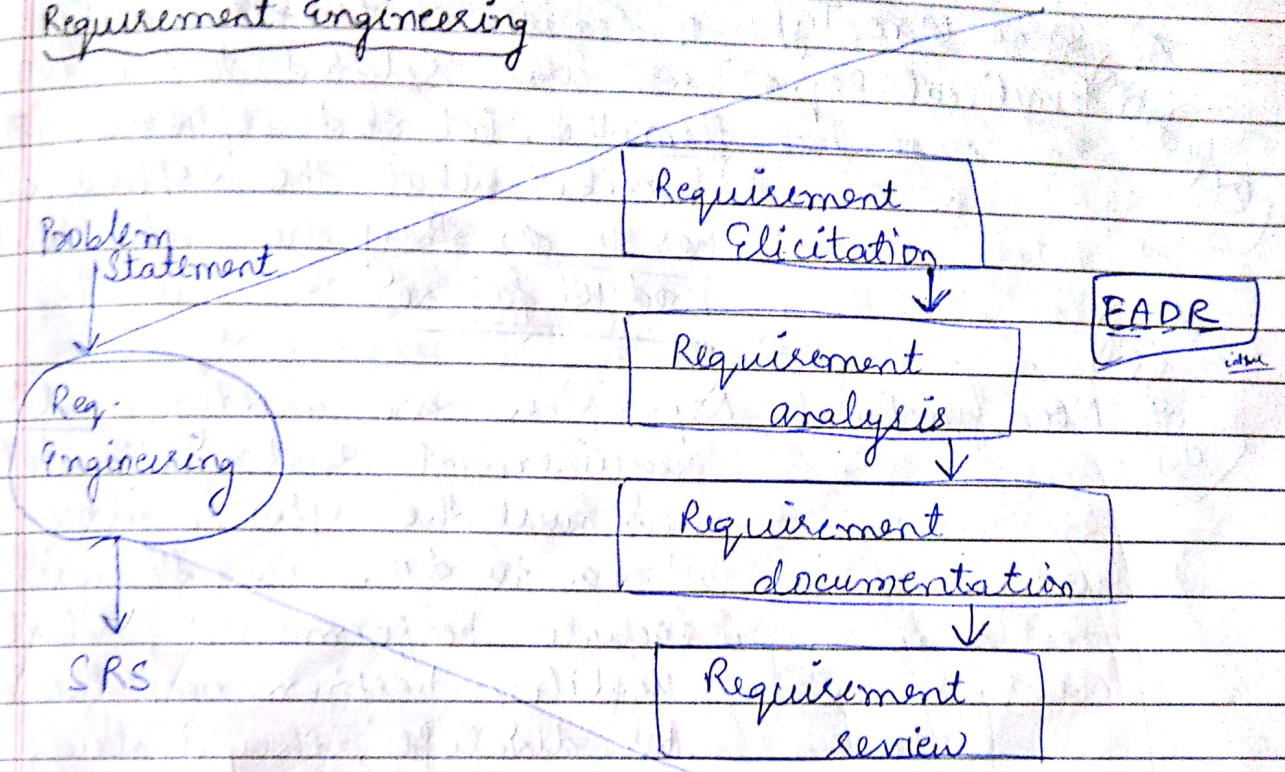
One phase is divided into 4 sectors:

1. Planning = determination of objective, alternative and constraint
2. Risk analysis = ~~analyse~~ analysis alternative & attempt to identify & resolve the risk involved.
3. Development = Product development & ~~testing~~ product.
4. Accessment = customer Evaluation

Date __/__/____ • Stake holders = directly or indirectly related to the system.

Page No.

Requirement Engineering



Type of Requirement:

1) Known Requirement = Something a stake holder believe to be implemented.

2) Unknown Requirement = Forgotten by stake holder because they are not needed right now or needed only by another stake holder.

3) Undreamt Requirement = Stake holder may not be able to think of new requirement due to limited domain knowledge.

"There are no great things, only small things with great love. Happy are those." Mother Teresa

Date __/__/____ Page No.: ____

☆) Some More Types of Requirements :-

(Expectation) Functional Req. = These are related to the expectation from the intended software. They describe what the software has to do. They are also called product feture.

(quality) Non-functional Req. = These are mostly quality requirements that stimulate how well the software does, what it has to do. For users, they include performance, reliability, usability, performance, flexibility. For developer, they include maintainability, portability, testability.

* STEPS OF REQUIREMENT ENGINEERING :-

1.→ Requirement Elicitation = (To gather the requirement).

a) Interview Technique.

. Selection of Stake holder

↦ Entry level personnal = Here, they may not have sufficient domain knowledge and experience but may be very useful for fresh ideas and different use

↦ Mid level Stakeholder = They have better domain knowledge & experience of project. They know the sensitive, complex and critical area of the project.

Date / /

Page No.:

↦ Manager or other stakeholder = Higher level management officers like vice president, General Manager, Managing director, ~~user of the s/w~~

→ Users of the software =

- Brainstorming Technique
  facilitator = resolve the conflict b/w team members
- FAST - Facilitated application specification Tech.
  This approach is similar to brainstorming session and the objective is to bridge the expectation gap i.e. a diff b/w what developers think they are suppose to build and what customers think they are going to get.
  Activities of fast session :-

→ Each participant presents his/her list of project, services, constraint and performance for discussion list may be displayed in the meeting by using board or large sheet of paper or any other mech. so that they are visible to all the participants.

→ The combine list for each topic are prepared by eliminating redundant entries and adding new ideas.

→ The combine list are again discussed and finalized by facilitator. Once the list has been completed the team is divided into further sub teams, each work'd to develop mini specification for one or more entries of the list.

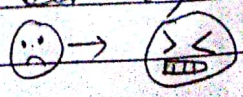
SRS → S/w Requirement specification.
Contract between client & server.
from SRS is signed before the project & if something is left, then SRS will be shown.

→ Each stakeholder then presents mini specification to all fast attendees. An
→ An issue list is prepared so that these ideas will be considered later.
→ Validation criteria is created.
→ Complete draft specification using all input from the fast meeting is created.

(Both) • QFD - quality function deployment.
It helps to encoorporate the voice of the customer. The voice is then translated into technical requirements.
→ Customer satisfaction is of prime concern and thus QFD emphasise on understanding of what is valuable to the customer & the deploy these values throughout the s/w engineering process.

* Three types of requirements in QFD-
→ Normal Requirement ✓
→ Expected ✓
→ Exciting ✓

Steps:-
1. Identify all the stakeholder.
2. List out the requirement from customer.
3. A value indicating a degree of importance. i.e. Priority is assigned to each of requirement.
5 Points : Very imp
4 " : imp
3 " : not imp but nice to have
2 " : not imp
1 " : Unrelastic (bilkul bekar)



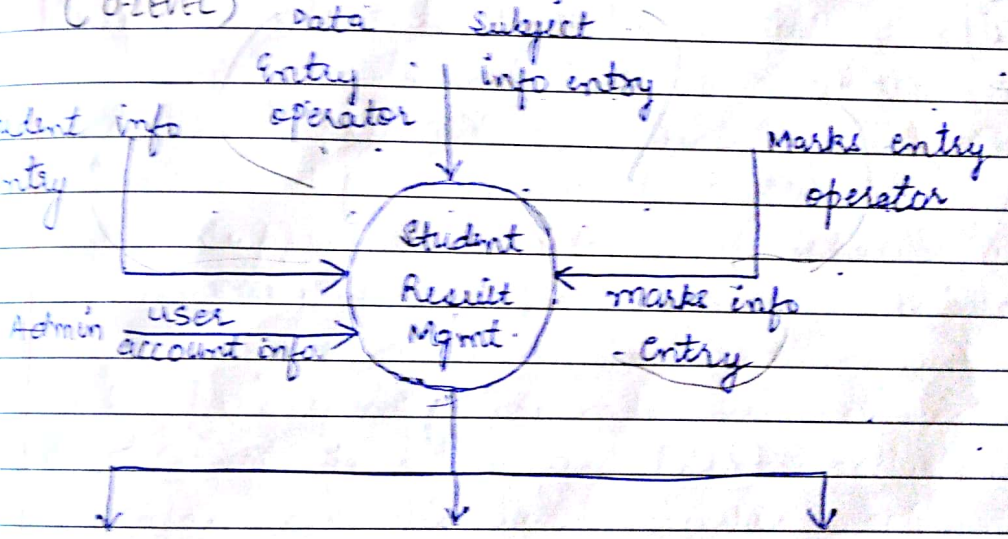
"Patience is a key element of success."-Bill Gates

* Data Modeling (ER Diagram)
* Flow Oriented Modeling

2. → Requirement Analysis

Draw the context diagram → Build the Prototype (optional) → Model the Requirement → Finalise the Requirement

CONTEXT DIAGRAM:
(0-LEVEL)



Student info Report generated

Marks sheet Generated

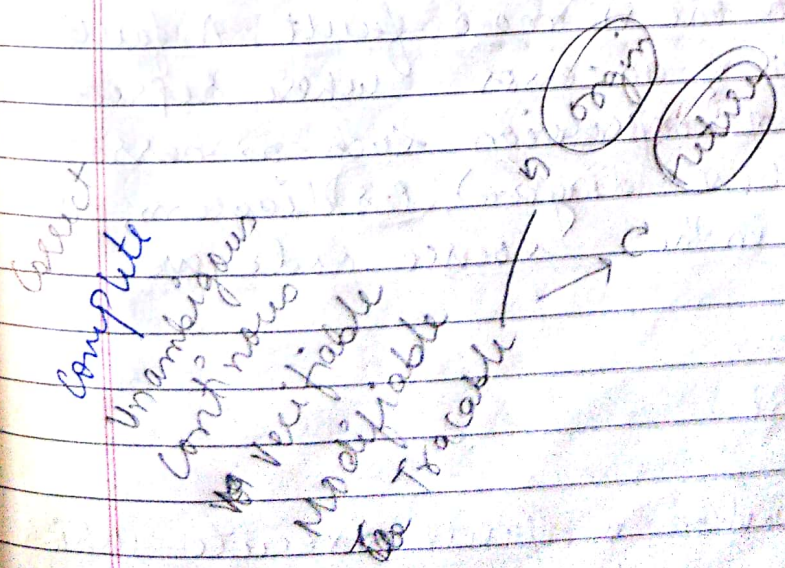
Student Performance report generated

Date __/__/____ Page No.:

* Characteristics of GOOD SRS:-

1. It should be correct.
2. It unambigous
3. Complete
4. Consistent
5. ~~Ranked for Importance~~
6. Verifiable
7. Modifiable
8. Traceable = a) Backword Tracing = Origin of requirement should be known

b) Forward Tracing = Future of requirement should be known.

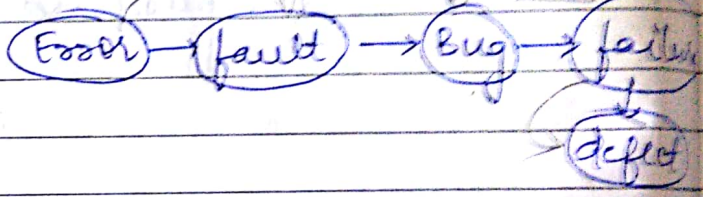


Date __/__/____ Page No.

chapter

SOFTWARE TESTING

People make (errors) synonym is mistake, this may be a syntax error or misunderstanding of requirement specification or logical error. Errors propagate from one phase to another with higher severity (impact). A requirement error may be magnified during design and amplified still more during coding. If it would not be detected before the release it may have serious implication in the field.



BUG

When developer make mistake while coding, we call these mistakes as bugs.

FAULT

An (error) may led to one or more fault. A fault is the representation of an error where representation is the mode of expression such as narrative (text), (dfd) (data flow diagram), (e-r) diagram, source code. If fault is in the source code we call it as bug.

FAILURE

A failure occurs when a fault execute which is the necessary and sufficient cond^n.

Date __/__/____

It is the departure of the output of the program from the expected output, hence failure is dynamic. The program has to execute for a failure to occur. A fault may lead to many failure.

DEFECT

It is said to be detected when a failure is observed by the users and developers.

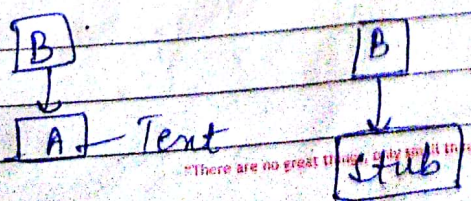
LEVELS OF TESTING

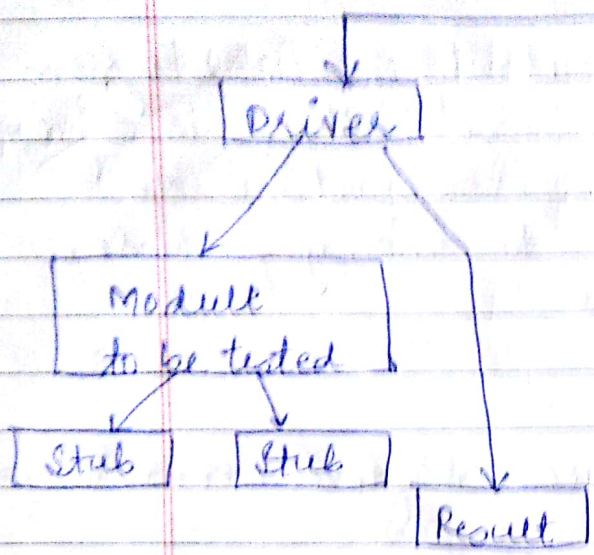
UNIT TESTING (concept of stub)

Example:

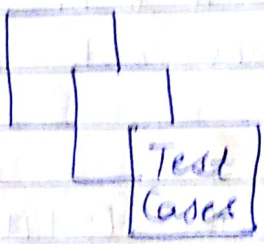
Suppose you have a fxn A that calculate the total marks obtained by a student in a particular academic year. Suppose this fxn derives its value from another fxn B which calculate the marks obtained in a particular subject. You have finished working on function A and want to test it. But the problem you face here is that you cant sum the fxn A without input from function B. Function B is still under development. This problem can be resolved in 3 ways:

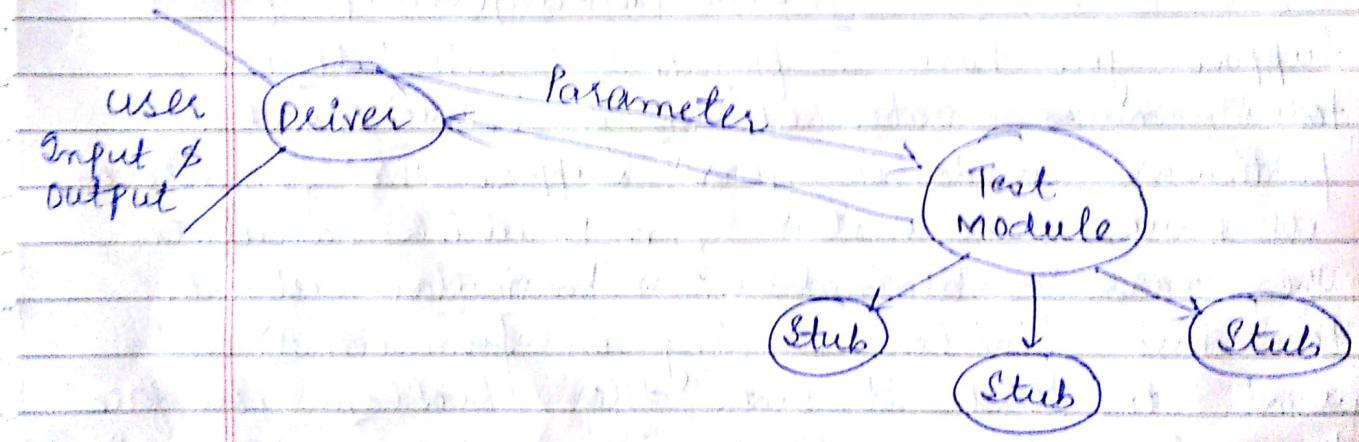
1. Complete testing can be postponed until the integration step.
2. Omit unit Testing and allow incremental addition.
3. Generate the Scaffolding.





Interface local data structure, Boundary conditions, Independent path, Independent path, error handling path





Date __/__/____

Page No. ____

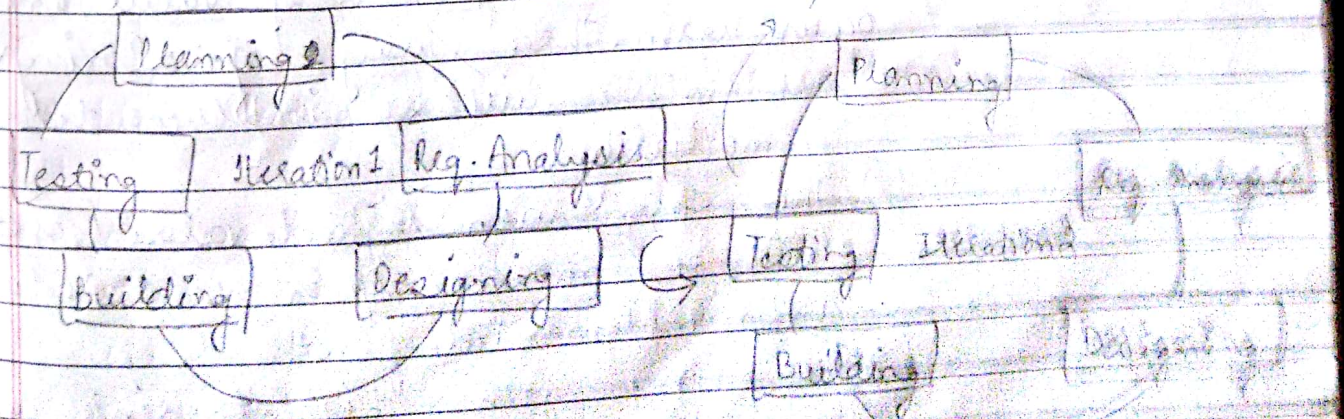
(10 marks) Imp

Agile

Agile SDLC (Software development life cycle) is a combination of iterative and incremental process model which focus on process adaptability and customer satisfaction by rapid delivery of working software product.

Agile method break the product into small incremental builds. These builds are coaged in iteration. Each iteration involve cross-functional team working simultaneously on various area like planning, requirement analysis, design coding, unit testing and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stake holder.

Agile model believes that every project need to be handled differently and the existing method need to be tailored to best suit of the project requirement. In agile, the tasks are divided into time boxed to deliver specific feature for a release. Iterative approach is taken in working software, build is delivered after each iteration.



"There are no great things, only small things with great love. Happy ... day" - Mother Teresa

Date __/__/____ Page No.: ____

Models of Agile:

A) XP (Xtreme Programming)

It has four phases = Planning → Designing → Coding → Testing.
(No req. analy.)

Phase 1 : PLANNING

Stories

→ User ~~Storage~~ = The planning activity begin with the creation of set of user ~~Storage~~ that describe required feature and functionality for software to be build.

→ Index Card = Each story (req.) is written by customer and is placed on the index card.
* assign priority or number.
* Member of the XP team access each story and assign a cost (measured in development weeks). If story require more than 3 development weeks, the customer is asked to split the story into smaller stories and assignment of value and cost occur again. Three way of odering stories:
(1) all stories will be implemented immediately.
(2) The story with highest value (cost) will be moved up in the schedule and implemented first.
(3) The story with highest risk will be implemented first.

"Patience is a key element of success." –Bill Gates

Page No.:

→ Compute project velocity = It is the number of customer stories implemented during the first release. It is used to: (1) Help estimate delivery date and schedule for subsequent releases. of next release
(2) Determine whether an over-commitment has been made for all stories across the entire development project.

Phase 2: DESIGN

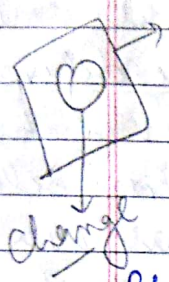
→ Keep the design as simple as possible.
→ CRC (Class responsibility Collaborator Card) = Class uses CRC which is a collection of standard index card that represent class.
→ Spike Solution = If a difficult design problem is encountered as part of the design on the story, XP recommend the immediate creation of an operational prototype of that portion of design called (SPIKE SOLUTION).

Phase 3: CODING

→ Pair Programming = Here we have a pair programmer.
→ Continous Integration = To avoid compatibility and integration problem. It provide (rough testing) smoke testing environment that help to uncover error early.

Date __/__/____ Page No.:

* Refactoring: It is a process of changing a software system in such a way that it does not alter the external behaviour of the code yet improve the internal structure. It minimises the chance of introducing bugs.



Phase 4: TESTING

* Unit Testing -
* Regression Testing = Testing of change.
* Acceptance Testing = Testing done by customer

AGILE VERSES TRADITIONAL SDLC MODEL

* Agile is based on the adaptive S/w development model whereas the Trad. SDLC Model like waterfall model is based upon predictive approach.
* In Traditional SDLC models, models usually work with detailed planning and have a complete forecast of the exact task and features to deliver in the next few months or during the product life cycle.
* Predictive method entirely depend on the requirement analysis and planning done in the beginning of life cycle.
* Any changes to be incorporated go through a strict change control management.
* Customer interaction is the backbone of agile methodology and ofen communication with

T

A

"Patience is a key element of success" - Bill Gates

Date __/__/____ Page No.: ____

Pros & Cons

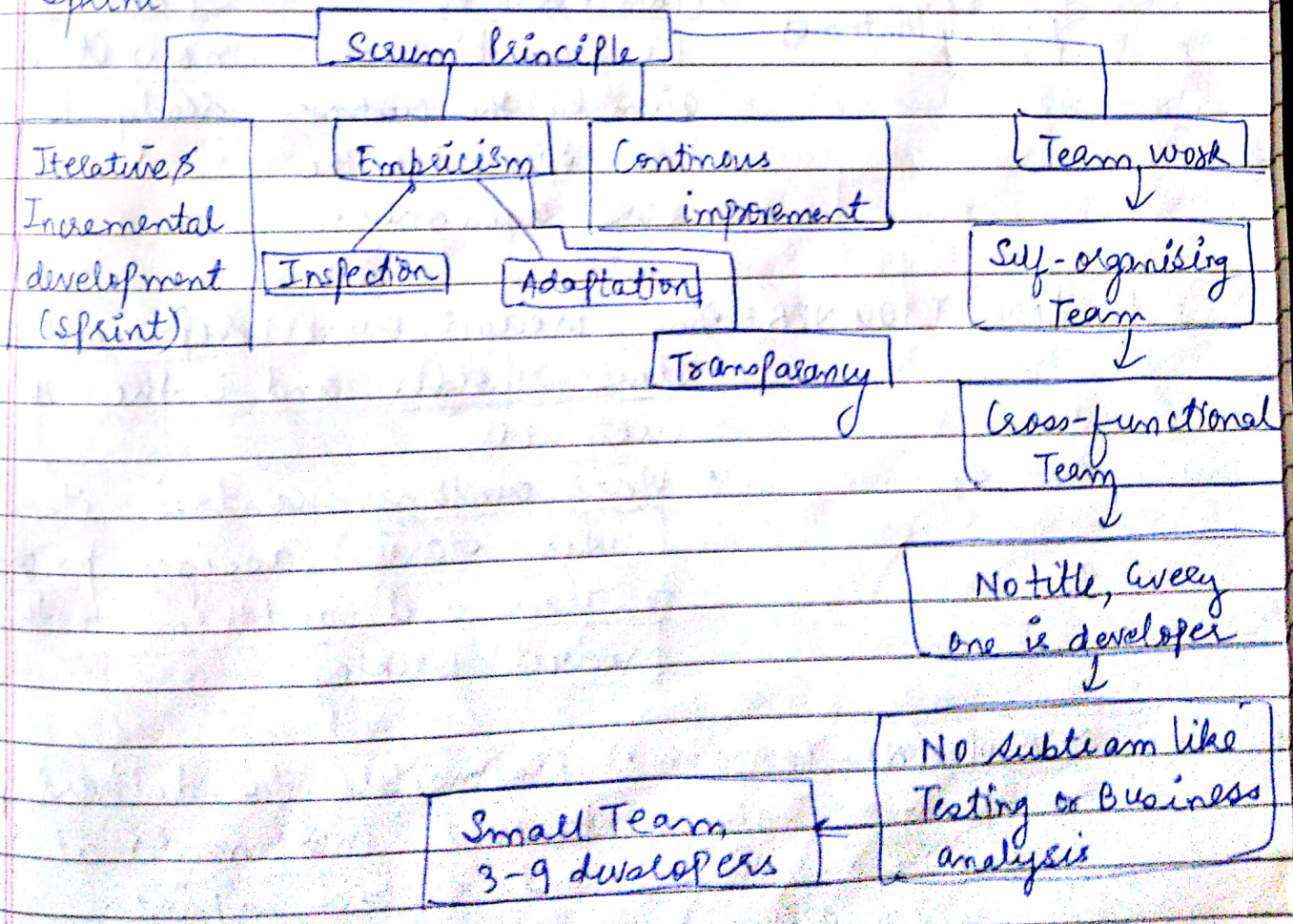
minimum documentation are the typical feature of agile development environment.

* The product is tested very frequently through the release iteration, minimising the risk of any failure in future.
* The agile team works in close collaboration with each other and most often located in the same geographical location.

b) SCRUM

Teams used =

1. Release log
2. Product backlog
3. Sprint



"There are no great things, only small things with great love. Happy are those." - Mother Teresa

Date __/__/____

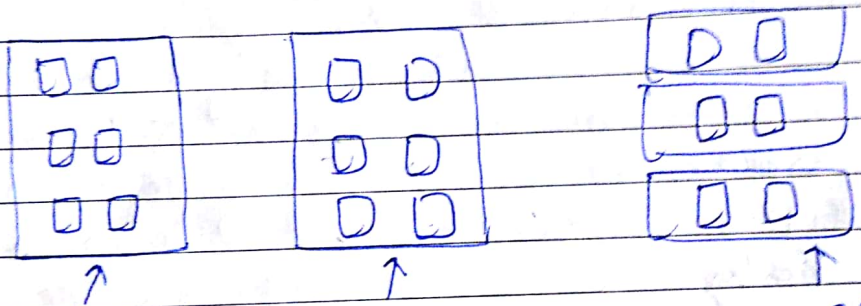
Page No.

→ SCRUM ROLE

PDSSM

1. Product Owner
2. Development Team
3. Scrum Master
4. Stakeholder
5. Managess

Product owner
Development Team
Scrum Master
Stake holder
Managers



requirement given by customes

Product backlog
The final product given by the ~~custom~~ developer with the cost assigned.

Sprint
now it is ready to release

1. PRODUCT OWNER = •) The product owner represent the stake holder and is the voice of customer.
   •) write ~~customer~~ centric items (user stories), assign priority to them and add them to the product backlog.

2. DEVELOPMENT TEAM = •) Responsible for delivering the sprint ke bolad jo bhi increments hai, unko deliver karta hai.

"Patience is a key element of success." ~Bill Gates

Date ___/___/______

Page No. ______

•) Having 3-9 people with cross functional skill team skills who do the actual work (analyse, design, develope, test document etc).

3. SCRUM MASTER =

•) It is not the team leader but act as a buffer b/w the team and any destracting influences.

•) Ensure that the scrum process is executed as intended.

•) Enforces the rule, protector of the team and keep it focused on the task

•) It differs from project manager.

TERMS → Jo customer ne mujhe requirements di hai, vo priority ke acc arrange krte hai backlog mai.

1. BACK LOG = •) The project requirement with assigned priority of feature that provide business value for to the customer.

•) Items can be added to the backlog at any time.

•) The product manager accesses the backlog and update the priorities as required.

2. SPRINT =

•) Consist of work unit that are required to achieve a requirement defined in the backlog that must fit into a predefined time box (Typically 30 days)

•) During the sprint, the backlog item on which the sprint works the req. are frozen (i.e. changes are not introduced during the sprint).

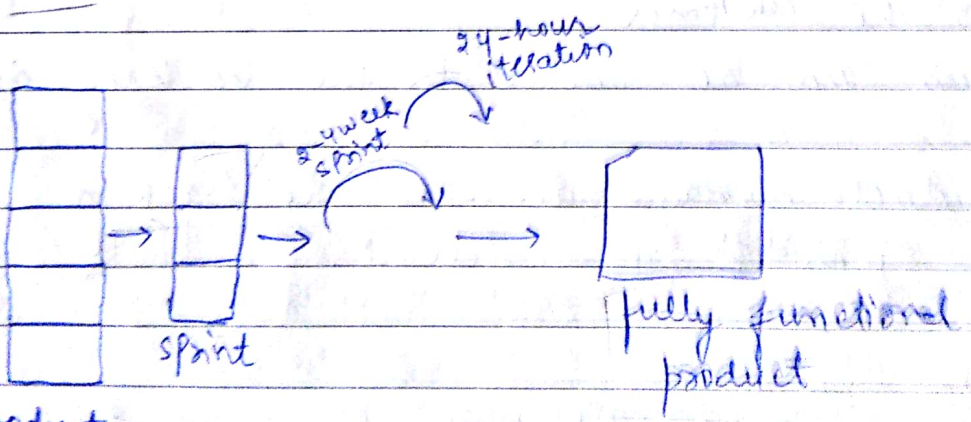
•) Hence sprint allow the team members to work in a short term but stable environment.

→ 8. SCRUM MEATINGS

These are short meatings (15 mins) held daily by Scrum team. 3 main questions are asked and answered by all team members.

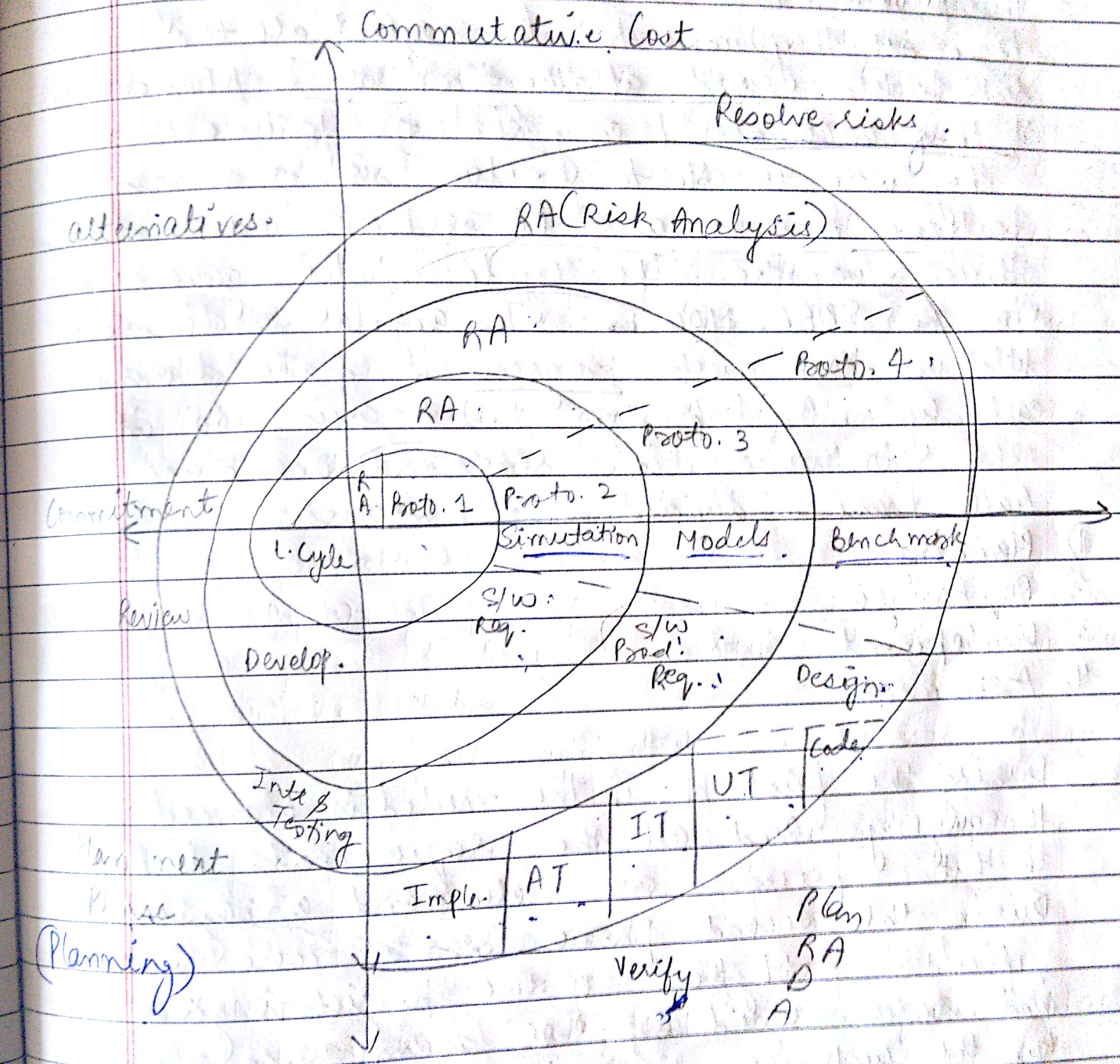
a) what did you do since the last team meating.
b) what obstacles are you encountering.
c) what do you plan to accomplish by the next team meating.

A team leader called a Scrum Master, leads the meating and accesses the responces from each person. The Scrum meatings help the team to uncover potential problem as early as possible.



Date __/__/____

Page No.: ____



Commitment

Partition

(Planning)

The Spiral

Date / / Page No.:

The previous models did not focus on uncertainty, hence all they used to do was neglect all the risk factors and at the time of implementation if they encountered any kind of failure then they were unable to solve it. So a new feature known as "PROJECT RISK" was introduced in the life cycle ∴ This gave rise to the SPIRAL MODEL. The angular dimension tells us how much progress is made during each cycle. A loop from the x-axis, 360° in the "clockwise dir" represents one phase. Each phase is divided into 4 sectors:

1) Planning
2) Req. Analysis
3) Development
4) Assessment

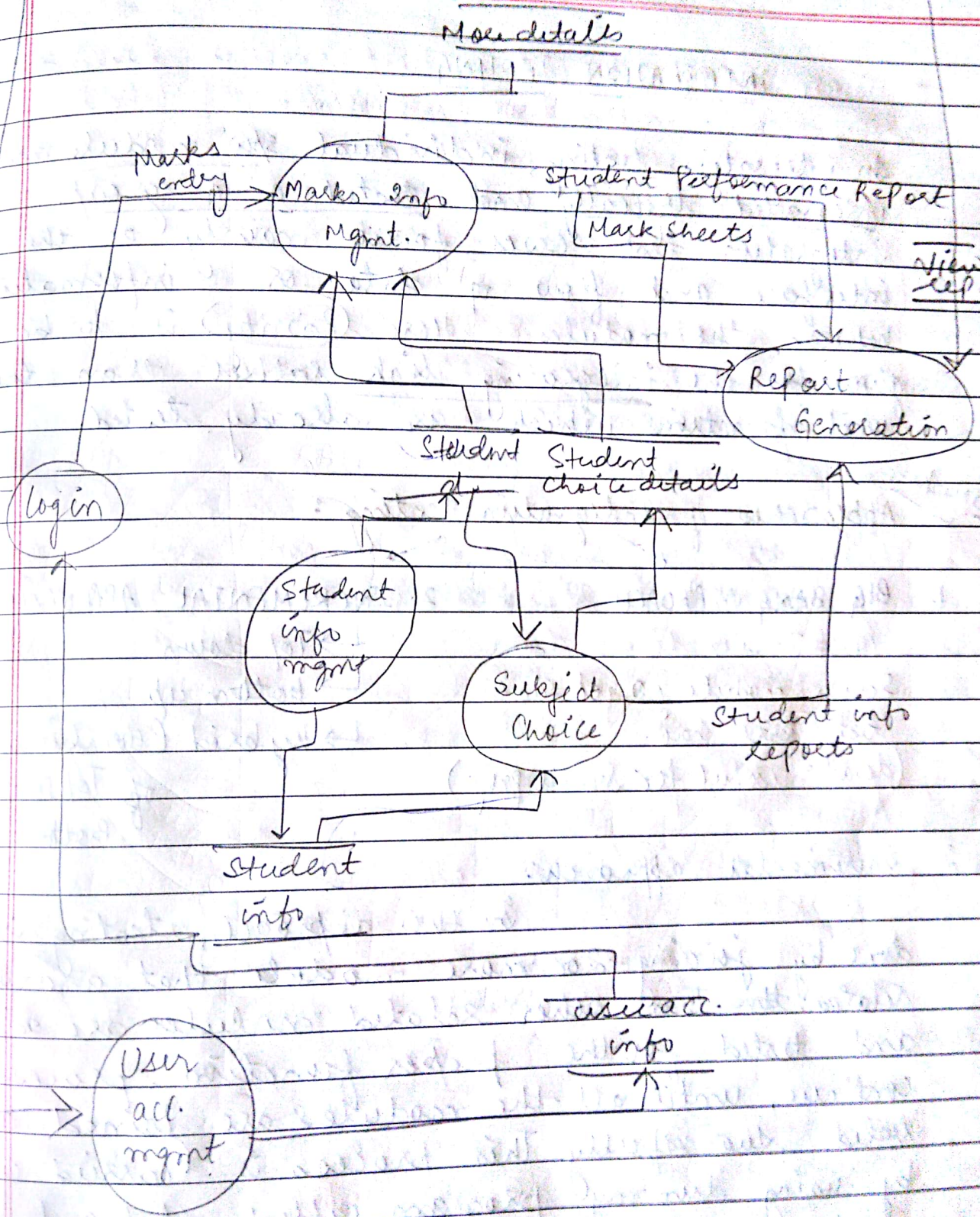
During the first phase, the model is planned thoroughly and all the requirements linked to that process are noted and analysed. During the second phase, a more refined prototype of the project is given and re-analysed or again assessed/checked by the customer. During the third phase all the risks have been analysed and a more good, effective s/w is developed.

The main imp. feature of the Spiral Model is that each phase requires the customer's assurance which is a + point.

Enter marks

Date __/__/____

Page No.: ____



Date __/__/__ ST-2 Page No.

INTEGRATION TESTING

In integration testing, individual s/w modules are integrated logically and tested as a group. Integration test cases, focus mainly on the interface and flow of data or information between the modules. Here Priority is to be given for the integrating link rather than the unit functions which are already tested.

10 Marks:

Approaches of Integration Testing :

1. BIG BANG APPROACH
(advantages & disadvantage)
Saare module ko ek saath test krte hai.
(disad - useful for small proj.)

2. INCREMENTAL APPROACH
↳ Top down
↳ Bottom up
↳ hybrid (combination of Top Down and Bottom up)

2. Incremental approach:

In this approach, testing is done by joining 2 or more module that are logical related. Then other related modules are added and tested for the proper function, process continue, until all the modules are joined and tested successfully. This process is carried out by using dummy program called stub and driver.

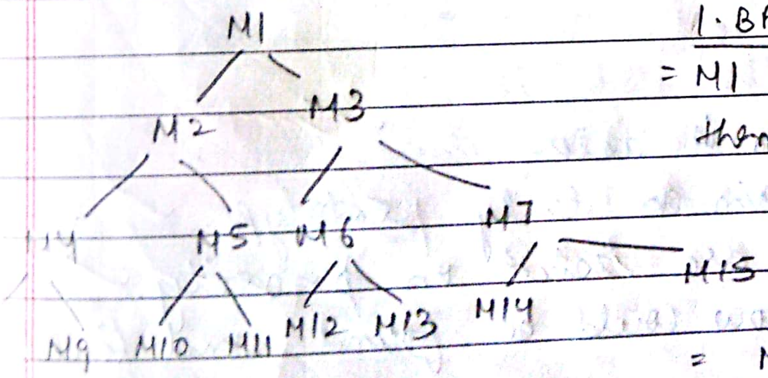
Stub and driver do not implement the entire programming logic of the software module but just simulate data comm. with the

calling module.

STUB: It is called by module under test, driver: ~~calles~~ calls the module to be tested.

3 approaches:

a) Top down integration: It is an incremental approach in which modules are integrated by moving downward ~~the~~ to the control flow, beginning with the main control module. 2 methods are used for integration: a) Depth first Search (DFS)
b) Breadth first Search (BFS)



1. BFS

= M1 then M2, M3 then M4, M5, M6, M7 then M8, M9, M10, M11, M12, M13, M14, M15

2. DFS

= M1 → M2 → M4 → M8 ~~...~~
then M1 → M2 → M4 → M9
then M1 → M2 → M5 → M10
then M1 → M2 → M5 → M11
and so on - - - -

Date __/__/____

b)

→ The integration process is performed in series of 5 steps. The main control module is used as test driver and stubs are substituded for all components directly subbordinate to the main control module.

→ Depending upon the integration approach selected (DFS & BFS), Subbordinate stubs are replaced one at a time with actual components.

→ Test are conducted as each component is integrated. On completion of each set of step, another stub is replaced with real component.

→ Regression Testing may be conducted to ensure that new errors have not been introduced.

Advantage:
1. Fault locatisation is easier.
2. Possible to obtain an early prototype.
3. Critical modules are tested on priority.
4. Major design flow could be found and fixed first.

Disadvantage:
1. Too many stubs are required.
2. Modules at lower ~~to~~ level are tested inadequately.
3. ~~Bottom~~

b) Bottom up Integration : In the bottom up approach, each module in lower level is tested with higher module until all module are tested. It takes help of drivers for testing.

Steps:

1. Low level components are combined into clusters that perform a specific software subfunction.
2. A drives is written to coordinate test case input and output.
3. The cluster is tested.
4. Drivers are removed and clusters are combined moving upward in the program structure.

Advantage:

1. Fault localisation is easier.
2. No time is wasted.
3. Waiting for all module to be developed unlike big bang approach.

Disadvantage:

1. Critical module are tested last / late and may be prone to defect.
2. Early prototype is not possible.

* WHITE BOX TESTING * (To study the internal structure of the program)

Cyclomatic Complexity

Defines the number of independent paths in the basic set of program and provides us with upper bound for the number of tests that must be conducted to ensure that all statements have been executed at least once.

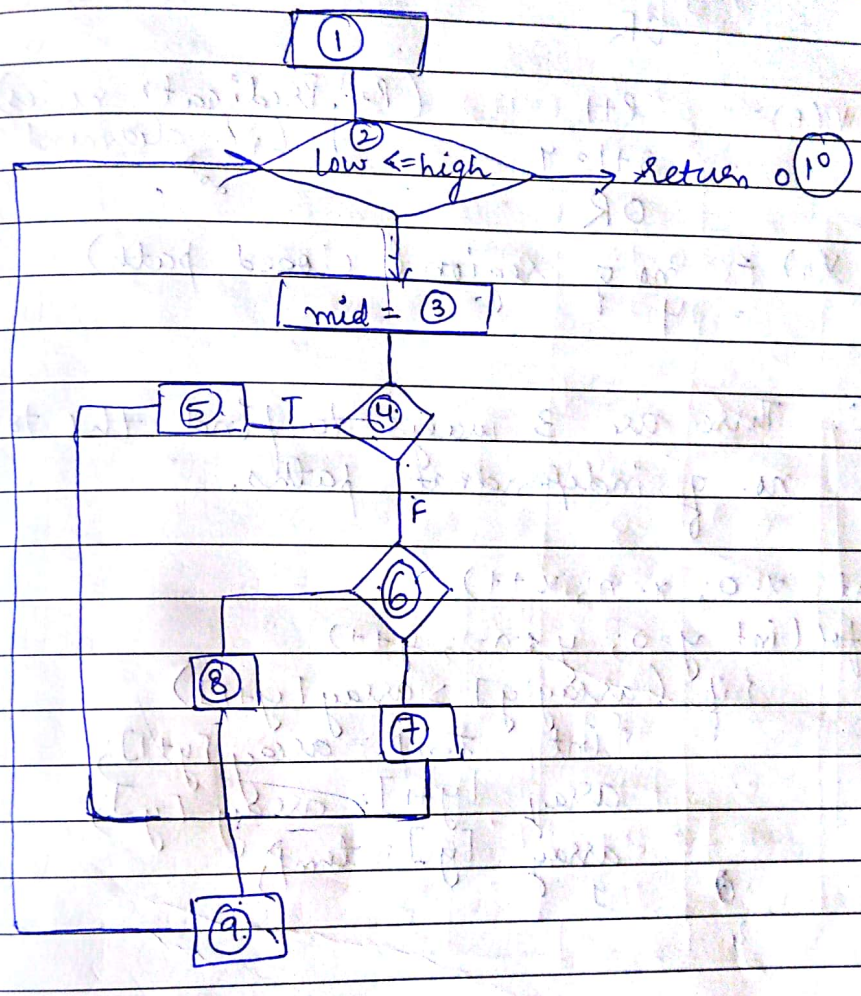
example of Binary Search:

```
int binSearch (int x, int v[], int n)
{ int low, high, mid;                    ]
  low = 0;                               ] —(1)
  high = n-1;                            ]
  while (low <= high)                    —(2)
    { mid = (low + high)/2;              —(3)
      if (x < v[mid])                    —(4)
        high = mid-1;                    —(5)
      else if (x > v[mid])               —(6)
        low = mid+1;                     —(7)
      else
        return mid;                      —(8)
    }                                    —(9)
  return 0;                              —(10)
}
```

Date __/__/____

Page No.: 6

∴ The flowchart corresponding to this code is:



Independent Path:

1. 1-2-10
2. 1-2-3-4-6-7-9-2-10
3. 1-2-3-4-6-8-9-2-10
4. 1-2-3-4-5-9-2-10

→ Path's through whole program is executed or traversed.

edges (i.e. lines which are joining)

$V(G) = E - N + 2 = 12 - 10 + 2 = 4$

(complexity) (nodes)

∴ 4 independent paths

OR

$V(G) = P + 1$ (P: Predicate nodes) (i.e. diamond symbols)

$= 3 + 1 = 4$

OR

$V(G) =$ no. of region (closed paths)

$= 4$

∴ There are 3 ways to find the total no. of independent paths.

Q

```
for(int x=0; x<n; x++)
    for(int y=0; y<n-1; y++)
        if (array[y] > array[y+1])
            {int temp = array[y+1];
            array[y+1] = array[y];
            array[y] = temp;
            }
    }
}
```

Ans - (c)

Date __/__/____

Page No. 10

* FUNCTIONAL TESTING *

① ~~Boundary Value analysis:~~

Functional Testing is testing which evolve only observation of the output. There is no attempt to analyse the code which produces the output ignoring the internal structure of the project.

→ (x, y) Range = 1-100

check x :) $(0,50), (1,50), (99,50), (100,50), (50,50)$

i.e. keep y constant & vary x.

check y :) $(50,0), (50,1), (50,99), (50,100)$

i.e. keep x constant & vary y

NOTE: In checking y, we will not check for $(50,50)$ as it is being done in x.

② Consider a program ~~an~~ consider nature of roots of quadratic eqn. Its input is a triple of +ve integer a,b,c and values may be from interval 0-100. The program output may have one of the following words:

1. ~~none~~ not a quadratic eqn.
2. imaginary root, equal root.
3. design the boundary, test cases

eqn: $ax^2 + bx + c$, at $(a,b,c) = (0-100)$

no. of test cases = $4n+1 = 13$

$= 4(3)+1$

(a,b,c)

"There are no great things, only small things with great love. Happy are those, Mother Teresa"

1801=

(0, 1, 99, 100, 50)

$b^2-4ac > 0$ = real roots
$b^2-4ac < 0$ = imag. roots
$b^2-4ac = 0$ = equal roots

$a = 0$ = not a quad eqn.

Test Case	a	b	c	O/P (Put these values in b^2-4ac.)
1	0	50	50	not a quad eqn.
2	1	50	50	real roots
3	50	50	50	imag roots
4	99	50	50	imag roots
5	100	50	50	imag
6	50	0	50	imag
7	50	1	50	imag
8	50	99	50	imag
9	50	100	50	equal
10	50	50	0	real
11	50	50	1	real
12	50	50	99	imag
13	50	50	100	imag

Q Consider a simple pgrm
Its input is a triple +ve integers say x, y, z and the data type for input parameter ensures that these will be integers greater than zero and less than equal to 100. The pgrm O/P may be one of the following work:

1. Scalene (All diff)
2. Isosceles (2 same)
3. Equilateral (all same)
4. Not a triangle (~~2 sides~~ sum of 2 sides, equal or more than 3rd)

"Patience is a key element of success." –Bill Gates

A

TestCase
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

Date / /

Page No. 12

A

(1, 2, 50, 99, 100)

~~4n+1 = 4(5)+1 = 20+1 = 21~~

4n+1 = 4(3)+1 = 12+1 = 13

Consider starting 2 values, ending 2 values and average since, more than 2 is 0 is not considered and 1, 2 are taken. If it would be equal to 0, then 0,1 would have been taken.

Testcase	a	b	c	o/p
1	1	50		
2	2	50		
3	50	50		
4	99	50		
5	100	50		
6	50	1		
7	50	2		
8	50	~~50~~ 99		
9	50	100		
10	50	~~100~~ 50	1	
11	50	50	2	
12	50	50	99	
13	50	50	100	
14				
15				
16				
17				
18				
19				
20				
21				

"There are no great things, only small things with great love. Happy are those." – Mother Teresa

Date

Robust meaning = JO unfavourable cond^n mai bhi chale.

Page No. 13

ROBUSTNESS TESTING

value ~~limit~~

It is extension of boundary ~~value~~ analysis. Here, we see what happens when extreme values are exceeded with the value slightly greater than the maximum and the value slightly less than the minimum.

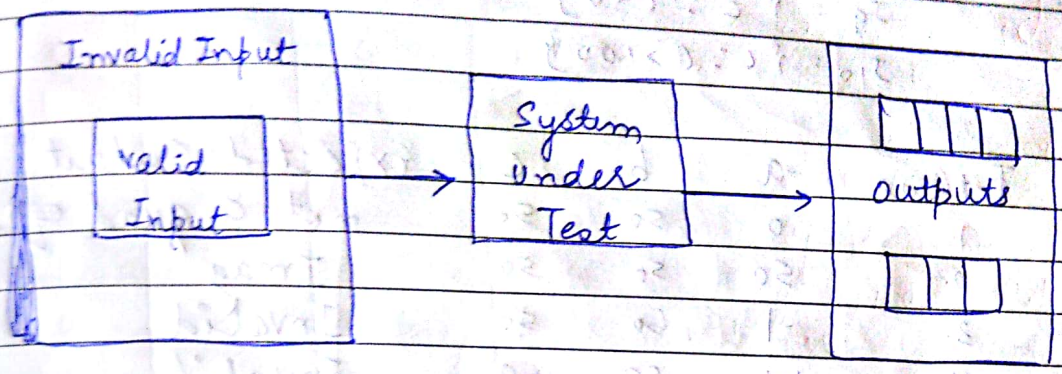
for (0-100)

Total no. of test cases = 6n+1.

-1, 0, 1, 50, 99, 100, 101

=) ~~Any value from valid & invalid range~~

EQUIVALANCE CLASS TESTING



01 { <a,b,c> : $b^2-4ac > 0$, real root)
02 { <a,b,c> : $b^2-4ac < 0$, imag root)
03 { <a,b,c> : $b^2-4ac = 0$, equal roots)
04 { <a,b,c> : a = 0; not a quad eqn) }

Test Case based on O/P

Test Case	a	b	c	Expected output.
1	0	50	50	not a quad eq.n.
2	1	50	50	real
3	50	50	50	imag
4	50	100	50	equal root.

range = (0 to 100)

$I_1 = \{a : a = 0\}$
$I_2 = \{a : 1 \le a \le 100\}$
$I_3 = \{a : a < 0\}$
$I_4 = \{a : a > 100\}$
$I_5 = \{b : 0 \le b \le 100\}$
$I_6 = \{b : b < 0\}$

$I_7 = \{ b : b > 100 \}$
$I_8 = \{ c : 0 \leq c \leq 100 \}$
$I_9 = \{ c : c < 0 \}$
$I_{10} = \{ c : c > 100 \}$

Test Case	a	b	c	Expected Output
1.	0	50	50	not a quad eqn.
2.	50	50	50	Imag
3.	-1	50	50	Invalid
4.	101	50	50	Invalid
5.	50	20	50	Invalid
6.	50	101	50	Invalid
7.	50	-1	50	Invalid
8.	50	50	20	Invalid
9.	50	50	-1	Invalid
10.	50	50	101	Invalid

Triangle

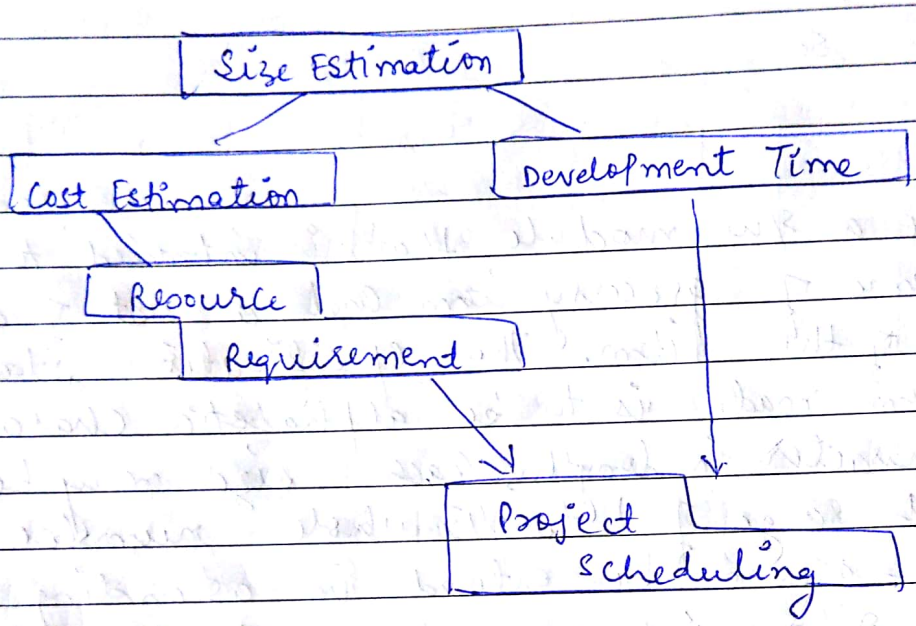
Test Case	a	b	c	O/P
1.	0	50	50	Invalid
2.	50	50	50	equal Δ
3.	-1	50	50	Invalid
4.	101	50	50	Invalid
5.	50	20	50	Isoceles
6.	50	101	50	Invalid
7.	50	-1	50	Invalid
8.	50	50	20	Isoceles
9.	50	50	-1	Invalid
10.	50	50	101	Invalid.

Date __/__/____

Test Case	a	b	c	expected output
1	1			
2	2			
3	So			
4				

Q. Consider a s/w module that is intended to accept the name of grocery item and a list of different sizes of the item. The specifications state that the item made is to be alphabetic character b/w 2-15 character in length, each size may be a value in the range of 1 to 48 whole number only. The size are to be entered in ascending order. A max of 5 sizes may be entered for each item. The item name is to be entered first, followed by a comma then followed by a list of sizes. A comma will be used to separate each size. Spaces are to be ignored anywhere in the input. Design the equivalance test cases.

Activities during software project planning



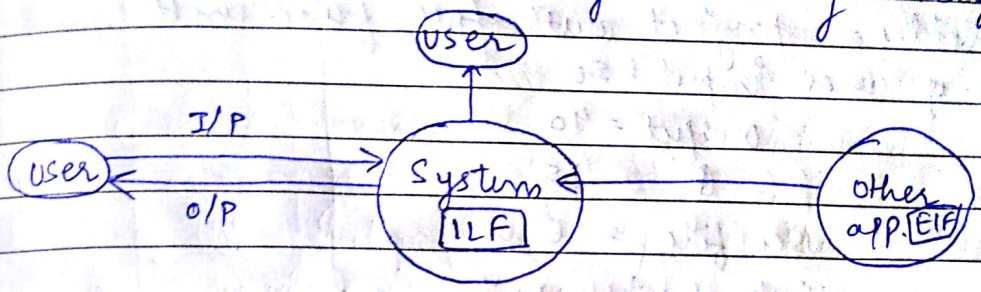
1. Size Estimation

a) Line of Code (LOC) = a line of code is any line of program text that is not a comment or blank line regardless of the number of statements or fragments of statements. This include all line containing program header, declaration, executable and non executable statements.

b) Function Count or Function Point = It measures functionality from the user point of view i.e. on the basis of what user request and recieve in return from the system. Function point analysis decomposes the system into 5 functional unit:

1. Input = Info. entering the System.
2. Output = " leaving " "
3. Enquiries = Request for instant access to information

(ILF) 4. Internal logical files = Info. held within the system.
(ELF) 5. External interface files = " " by the other system that is used by the system being analysed.



Numerical:

Learn this table:-

functional unit	weighing factor		
	low	avg	high
I	3	4	6
O	4	5	7
Q	3	4	6
ILF	7	10	15
ELF	5	7	10

$$UFP = \sum_{i=1}^{5} \sum_{j=1}^{3} z_{ij} w_{ij}$$

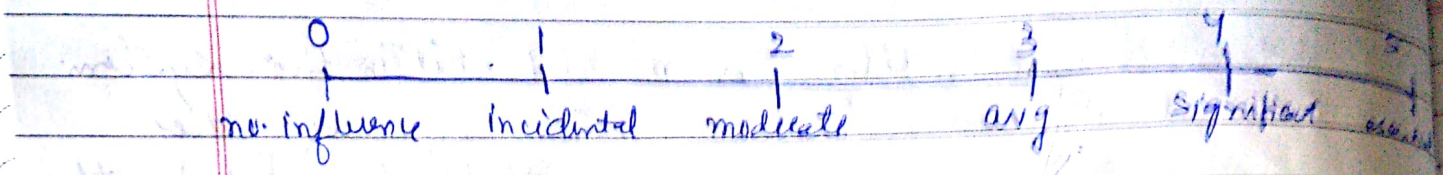
↓ Unadjusted function point

$$FP = UFP * CAF \text{ (complexity adjustment factor)}$$

↓ function Point

$$CAF = \left[0.65 + 0.01 \times \sum_{i=1}^{14} f_i \right]$$

"There are no great things, only small things with great love. Happy are those, who matter loves"



Q Consider a project with fol. func. unit.
no. of user input = 50
output = 40
@ = 35
user file = 6
external interface = 4
Assume all complexity adjustment factor & waiting factor all avg. Compute the fxn point of project.

A
$$UFP = 50\times4 + 40\times5 + 35\times4 + 6\times10 + 4\times7$$
$$= 628$$

(avg. value acc. to ques from scale.)

$$CAF = 0.65 + 0.01 \times (14\times3)$$
$$= 1.07$$

$$FP = 628 \times 1.07 = 672 \text{ (approx)}$$

Q An application has the following 10 low ext I/P
12 high ext. O/P
20 low ext. logical files
15 high ext int files
12 avg enquiries
& val. of complexity adj. factor = 1.10 - what are the adjusted & unadjusted FP.
i.e. FP, UFP = ?

Date __/__/____

A⇒

	I	10
	O	12
	Q	12
ILF	[redacted]	20
	EIF	15

UFP = 10×3 + 12×7 + 12×4 + 20×[redacted]7 + 15×10

= [redacted]452

FP = UFP * CAF

= 452 × 1.10

= 497.2 Ans

Q⇒ Consider a proj. with para

Ext I/P

10 with low complexity

15 " avg "

17 " high "

Ext O/P

6 with low complexity

13 " high "

Ext enquiries

3 with low "

4 with avg "

2 " high "

(ILF) 1 logical file

2 with avg "

1 " high "

(EIF) 1 interface file

9 with low "

imp

In addition to above system require:

1. Significant data comm.
2. Performance is very critical. [i.e. very critical]
3. Design code may be moderate
4. System is not designed for multiple installation in diff organisation.

Other complexity adjustment are treated as avg Compute FP for the project.

A

$$\Sigma f_i = 4+5+2+0+(10\times3)$$
$$= 11+30 = 41$$
$$CAF = 0.65 + 0.01 \times 41$$
$$= 1.06$$
$$FP = UFP \times CAF$$
$$= 424 \times 1.06 = 449.44$$

To calculate Numericals for Matrix:

Learn

$$\text{Productivity} = \frac{FP}{\text{Person-month}}$$

$$\text{Quality} = \frac{\text{Defects}}{FP}$$

$$\text{Cost} = \frac{\text{Rupees}}{FP}$$

$$\text{Documentation} = \frac{\text{Pages of documentation}}{FP}$$

* Table for Basic Model

Project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Date / /

Page No.: 22

Constructive Cost Model (cocomo)

B.W. Boehm - proposed the cocomo model.

Two Types:

cocomo - I — Basic / Intermediate / Detailed

cocomo - II

Basic Model: quick and rough Estimation.

a) Effort (E) = $a_b (KLOC)^{b_b}$ Person-Month [KLOC = K-Line of Code / Kilo]

b) Development Time (D) = ~~E_b Effort~~ $C_b (Effort)^{d_b}$ Month

Table:

Mode	Project Size	Nature of Project	Innovation	Deadline of the Project	Development environment
Organic	2-50 KLOC	Little		Not tight	Familiarity in house
Semi-detached	50 - 300 KLOC	Medium		Medium	Medium
Embedded	over 300 KLOC	Significant		Tight	Complex H/W / customer interface Requirem

"There are no great things, only small things with great love. Happy are those who gather here"

~~Numerical~~

c) Average Staff Size = $\frac{\text{Effort}}{\text{development time (unit = months)}}$

Its unit is Person.

d) Productivity = $\frac{\text{KLOC}}{\text{PM (Person Month)}}$

Numerical:

Q. Suppose that a project was estimated to be 400 KLOC. Calculate the effort and development time for each of the 3 modes i.e. organic, semi detached, embedded.

A. ORGANIC Effort = $a_b (KLOC)^{b_b}$ Person-Month

$= 2.4 (400)^{1.05}$

$= 1295.311$ Person-Month

development Time = $C_b (\text{Effort})^{d_b}$

$= 2.5 (1295.311)^{0.38}$

$= 38.075$ Month

SEMI-DETACHED

Effort = $a_b (KLOC)^{b_b}$

$= 3.0 (400)^{1.12}$

$= 2462.796$

develop

Date __/__/____

Page No.: 24

Q. A project size is of 200 KLOC is to be developed, software development team has average experience on similar type of project. The project schedule is not very tight. Calculate the effort, d.T, avg staff size, productivity of project. (Development Time)

A

Date __/__/____ Page No.: ____

MATRIX

(1) PRODUCT MATRIX

Describe the characteristics of the Product such as size, complexity, design feature, performance, efficiency, reliability, portability etc.

Describe the effectiveness & quality of produce that pro

time to produce the product, effectiveness of defect removal during development, no. of defect found during testing.

Maturity of the Process:

(2) PROJECT MATRIX

Describe the project characteristics and execution.
eg → no. of s/w developers, staffing pattern over the life cycle of the software, Cost and schedule, Productivity.

There are 2 measures of s/w Process (cost and effort applied) and product (LOC, execution speed, defect reported over some set period of time). Indirect mesure all the product that include functionality, quality, complexity, efficiency, reliability, maintenability and many other.

Date __/__/____

Page No.: ____

Q.5 OBJECT ORIENTED MATRIX

1) Number of Scenario Script
A Scenario script is a detailed sequence of steps that describe the interaction b/w the user and application.

2) Number of Key Classes
These are the independent classes (Base class).

3) Number of Support classes
Total no. of independent class

4) Avg. no of Support classes per class
ek key class ke saath kitni dependency h.

5) Number of Subsystem
A subsystem is an aggregation of classes that support a fxn that is visible to the end user of a system

6) Matrix for Software Quality
(A) Correctness: It is the degree to which the software perform its required function. Measured in Defect/KLOC.

(B) Maintainability: It is the ease with which a program can be corrected if an error is encountered, adapted if its environment changes or enhanced if the customer desire a change in the requirement.

"There are no great things, only small things with great love. Happy are those." - Mother Teresa

Date __/__/____ Page No.:

correct (when an error) | adapt (when environment change) | enhance

*) It is measured in MTTC (Mean Time to change)
Mean Time to change is a time oriented matrix that calculate the time to analyse the change request, design an appropriate modification, implement the change, test it and distributed the change to all user.

(C) Integrity: This attribute measure a system's ability to withstand a tax (both accidental and intentional) to its security. To measure integrity to additional attribute must be defined:

(1) THREAT: It is the probability that an attack of a specific type will occur within a given time.

(2) SECURITY: It is the probability that the attack of a specific type will be repelled.

$$\text{Integrity} = \Sigma[1 - (\text{threat} \times (1 - \text{security}))]$$

Date __/__/____ Page No.

(D) Usability : If a program is not easy to use, it is often lead to failure. Even if the function that it performs are valuable.

(E) Defect removal efficiency : It is measure of the filtering ability of quality assurance and control activities as they are applied throughout all process framework activities.

(DRE)

$$DRE = \frac{E}{E+D}$$

E = no. of error found before delivery of the s/w to the end user.

D = no. of defect found after delivery.

$$DRE = \frac{E_i}{E_i + E_{i+1}}$$

Imp

EARNED VALUE ANALYSIS

1. Budget Cost of Work Schedule (BCWS) or PV (Planned value).
2. Budget Cost of work performed (BCWP) or Gained Value.
3. Budget at Completion (BAC)
4. Actual Cost of work Performed (ACWP) or AC (Actual Cost)
5. SPI (Scheduled Performance index) $= \frac{BCWP}{BCWS} = \frac{EV}{PV}$
6. Scheduled Variance (SV) = BCWP − BCWS
7. CPI (Cost Performance Index) = $\frac{BCWP}{ACWP}$, $CPI = \frac{EV}{AC}$ (Earned Value ↑, AC ↓ actual cost)
8. Cost Variance (CV) = BCWP − ACWP

"There are no great things, only small things with great love. Happy are those." - Mother Teresa

11. PV = Planned completion (%) * BAC

12. EV = Actual completion (%) * BAC

Date / / Page No.:

13. $SPI = \frac{EV}{PV}$, 14. $\frac{EV}{AC} = CPI$

9. Scheduled for Completion = $\frac{BCWS}{BAC}$

10. % Complete = $\frac{BCWP}{BAC}$

SPI (Scheduled Performance Index)
Represent how close actual work is being completed, compared to Scheduled.

If 1 = then correct.
If less than 1 = then we are lacking behind
If more than 1 = then we are working ahead

A value of above 1 means that the Project is doing well against the schedule.

Ques Q Suppose Assume that you are a Software product manager and that you have been asked to compute earned value for a small software project. The project has 56 planned work task that are estimated to require 582 Person-days to complete. At the time you have been asked to do the earned value, 12 task have been completed however the project schedule indicate that 15 tasks should have been completed. The following Scheduling data in Person-days is given:

(P.T.O)

Date __/__/____

Page No.: _____

Task	Planned effort	Actual effort
1.	12.0	12.5
2.	15.0	11
3.	13.0	17.0
4.	8.0	9.5
5.	9.5	~~10.0~~ 9.0
6.	18.0	~~10.0~~ 19.0
7.	10.0	~~11.5~~ ~~10~~ 10.0
8.	4.0	~~10.5~~ 4.5
9.	12.0	~~6.5~~ 10.5
10.	6.0	~~4.0~~ 6.5
11.	5.0	~~14~~ 4.0
12.	14.0	14
13.	16.0	—
14.	6.0	—
15.	8.00	—

BCWS = 156.5

BCWP = of 12 (Performed Tasks)

= 12.0 + 15.0 - - - - - - 14.0

= 129.5

SPI = 80.8 Person

SV = -30

CPI = 99.2%

"There are no great things, only small things with great love. Happy are those." - Mother Teresa

Date __/__/____ Page No:

Q Suppose you have a budget cost of a Project at 9 lack dollar. The Project is to be completed in 9 months. After a month you have completed 10% of the Project at a total expense of 1 lakh dollar. The planned completion should have been 15%. Check Project is in budget & on time
(So calculate SPI & CPI).
Less, more, equal to 1 wala concept.

A

Planned Completion = 15%
BAC = 9 lack
Budged Cost = 15% of 9 lakh.
PV = 15% * 9 = 1,35,000
BCWP = 10% of BAC
EV = 90,000

$$SPI = \frac{90,000}{1,35,000}$$ Ans.

$$CPI = \frac{90,000}{1,00,000}$$ Ans.

Q Suppose you are managing a S/W development proj. The proj is expected to be completed in 8 months at a cost of 10,000 dollar per month. After 2 months you reliase that Project is 30% completed at a cost of 40,000 dollar. You need to determine weather the project is on time and on budget after 2 months.

A

BAC = 80,000 (8 × 10,000)
AC = 40,000
Planned Completion = 2/8 = 25%.
Actual = 30%.

"Patience is a key element of success."–Bill Gates

Date __/__/____

Page No.: ____

PV = 25% of 80,000
= 20,000

EV = 30% of 80,000
= 24,000

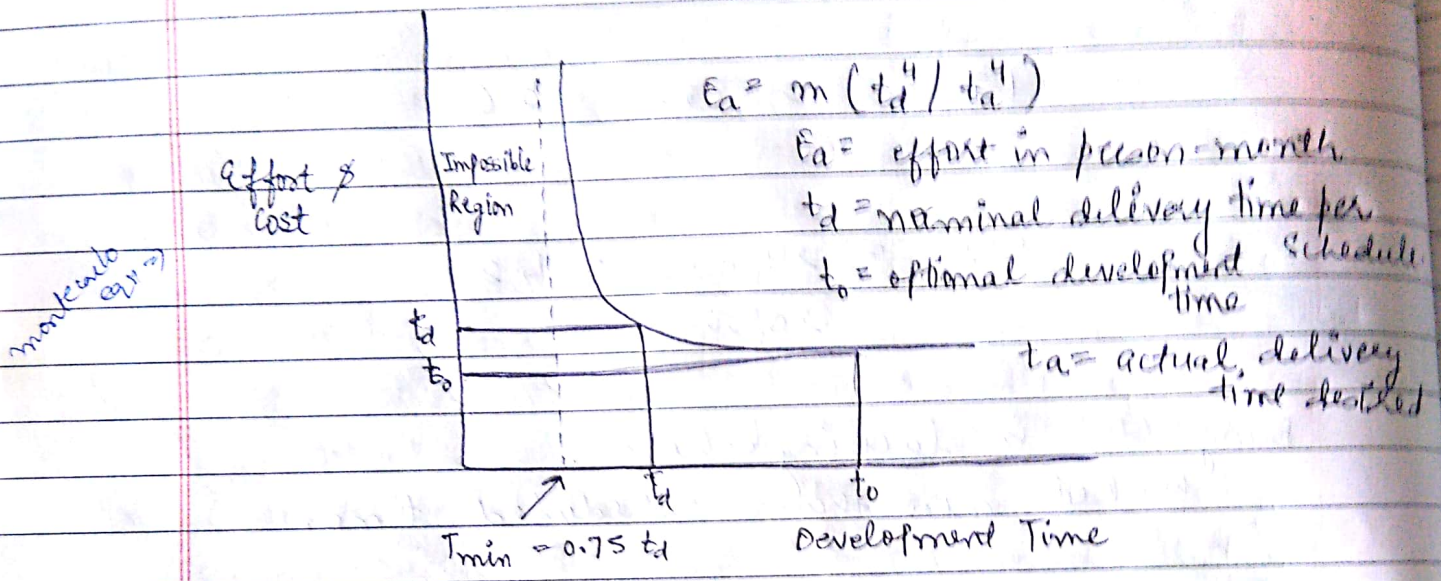
CPI = $\frac{24,000}{} = 0.6$

SPI = $\frac{24,000}{20,000} = 1.2$

Budget ~~cost~~ is lacking behind.
worked more than scheduled time.

Date __/__/__ Page No.:

Relationship b/w People and effort



$E_a = m(t_d^4 / t_a^4)$

E_a = effort in person-month

t_d = nominal delivery time per schedule

t_0 = optimal developed time

t_a = actual delivery time desired

Putnam-Norden-Rayleigh (PNR) Curve

Expected value for the estimation variable (size) (S)

$$S = (S_{opt} + 4S_m + S_{pess})/6$$

S_{opt} = optimal size

most likely = (S_m)

Pessimistic = (S_{pess})

Date __/__/____

SYSTEM TESTING

(A) RECOVERY TESTING : MTTR: mean time to recover

(B) SECURITY TESTING :

(C) STRESS TESTING :

(D) PERFORMANCE TESTING :

Date __/__/____ Page No.:

VALIDATION TESTING

(A) ~~acceptance testing~~

(B) α-Testing β-Testing

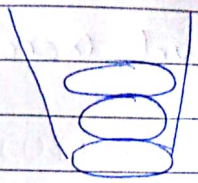
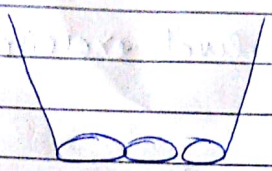
	α-Testing	β-Testing
1.	It is conducted at the developer side by customer or developer that can be controlled by only developer.	1. It is always performed by the customer at their own site but can not be controlled by the developer.
2.	It involve both whitebox & black box testing.	2. It involves only black box testing.
3.	Long execution cycle required.	3. Only few week of execution.
4.	Critical issue or fault can be addressed by developer's immediately in α-Testing.	4. Most of the issues or feedback is collected from β-Testing, will be implemented in future version of the project.
5.	Reliability & Security are not performed in depth.	5. Reliability, security, Robustness are checked during β-Testing

"Patience is a key element of success." – Bill Gates

Date ___/___/_____

Page No.: _____

Verification & Validation

Verification	Validation
1. eg→ Are we built the product right.	1. eg→ Are we build the right product
	
2. It is the process of evaluating a system or component to determine weather the product with a given development phase, satisfy the conditions imposed at the start of that phase.	2. It is the process of evaluating a system or component during or at the end of the development process to determine
3. Verification is a static Practice of verifying document design, code and program.	3. Validation is a dynamic mechanism of validating and testing the actual product.
4. It is human based checking of document and files.	4. It is computer based execution of program.
5. It generally comes first before validation.	5. It follows after verifi.
6. It uses method's like inspection, review, walk through & desk checking.	6. It uses method like blackbox, white box testing.

"There are no great things, only small things with great love. Happy are those."—Mother Teresa

Date __/__/____ Page No.: ____

7. Target is application & S/W architecture design and database design.	7. Target is the actual product like a unit/ module, integrated module & effective final product.
8. High level execise	8. Low level execise

→

BLACK BOX TESTING (functional Testing)	WHITE BOX TESTING
1. It is the S/w testing method, which is used to test the software without knowing the internal structure of the code or program.	1. S/w testing method in which internal structure is known to tester
2. System, acceptance Testing. (High level)	2. Unit, Intlgration (Low level)
3. Concentrate on the functionality of the system.	3. Concentrate on testing of progeam code of the system like code structure, branches, cond^n, loops etc.

Date __/__/_____

4. The main aim of the testing is to check on what functionality is performing by the system under test.	4. The main aim of this is to check on how system is performing.

SRS - S/W requirement specification [requirements r given]
SDD - S/W design description. [detailed design is given to developer].

5. eg → SRS (Based on)	5. SDD → eg. (Based on)