

Q.No.											Q.No.
-------	--	--	--	--	--	--	--	--	--	--	-------

प्र.कृ.
Q.No.

Name

• Sarang G.

Deodhar

• SDM

• Prof :- Gandhi Sir

Q.No.

Q.No.

प्र.क्र.
Q.No.

Day 1

Introduction

Q Why do we need Engineering?

=> To get product having,

① Better Quality

② Better Price

③ Better turnaround times.

Q What is Quality?

=> ① Meeting the specifications

② Meeting the requirements

③ Fitness for use

④ Meeting the stated, implied and legal requirements.

⑤ Minimum loss to the society.

Q How do we achieve Quality?

=> ① Inspecting & Testing - Quality control.

② Reviews and verification - Quality Assurance.

③ Process Approach (kind of process).

④ TQM (Total quality management)

- requires constant improvement

- If you are not moving forward, you are moving backward.

⑤ Quality by design (Systematic approach)

→ predefined objective.

→ product & process understanding

→ process based on

Q.No.

Q.No.

Q.No.

→ process control based on
 ① sound science
 ② quality risk management

⑥ six sigma

- Define (identify & define poor function areas)
- Measure
- Analyze
- Improve
- Innovate
- Control.

Day 2 - SDLC

* Myths :-

① Management

- We have a book containing all standards...
- If we are behind the schedule, we can add more programmers and catch up.
- We can always decide to outsource if we do not know how to build.

② Customer

- Basic requirements should be good enough to start coding. We can add more details of what we want later.
- We agree that requirements changes continuously, so what? change can be accommodated since software is flexible.

③ Programmer

- Once we done with the program our job is done.

Q.No.

Q.No.

प्र.क्र.
Q.No.

→ Until I finish coding, how can I assess the quality?

→ The only deliverable work product is a working program.

→ Software engineering practices make us write more documents and slows down our work.

Q What is process?

- ⇒ ① A good process can deliver good output
- ② Engineering,
- ③ It is a software development life cycle

* SDLC

- ⇒ ① Requirements / Analysis [owner, man]
- ② Design
- ③

* SDLC Phases :

- ⇒ ① Requirements / Analysis
 - product owner
 - product manager
 - Business Analyst / Product Analyst
 - CTO

② Design

- System Architect
- UX/UI design

③ Development

- Front-End developer
- Back-End developer

Q.No.

Q.No.

ঃ ক.
Q.No.

④ Testing

- Solutions Architect
- QA Engineer
- Testers
- DevOps

⑤ Deployment

- Data Administrator
- DevOps

⑥ Maintenance.

- Users
- Testers
- Support managers.

Q. What is SDLC?

- ⇒ ① SDLC is a framework containing the processes, activities and the tasks involved in the development, operation and maintenance.
- ② It also helps to decide spanning the life of the system from definition of its requirements to the termination of its use.
- (ISO/IEC 12207)

* SDLC Models

- ⇒ ① Waterfall model
- ② Spiral model
- ③ Prototype model
- ④ Iterative model
- ⑤ Rapid application development model

Q.No.

Q.No.

प्र.क्र.
Q.No.

Q

What is Waterfall Model?

- (1) It is a sequential model that divides S/w development into predefined phases.
- (2) Each phase should be completed before the next phase can begin with no overlap between the phases.
- (3) Review & approval of each phase outputs.
- (4) Model does not permit going back & forth.
- (5) If any defect found, go back to the originating phase & start traversing sequentially all over again.

* Advantages:-

- (1) projects under control
- (2) Pre-defined outputs at every phase
- (3) Tracking changes is easy.
- (4) Early identification of slippages, if any

* Disadvantages:-

- (1) In real life, customer requirements do change.
- (2) Customer appraisal of completed work - not feasible always.
- (3) Phases can not run concurrently.

Q.No.

Q.No.

प्र.क्र.
Q.No.

① Spiral Model?

- ⇒ ① In Spiral Model, software is developed in a series of incremental release.
- ② The spiral Model is divided into a number of framework activities or task regions
- ③ suitable for large projects with multi-location implementation.
- ④ Each spiral consists of deliverable product.
- ⑤ Feedback of each spiral is incorporated in next spiral.
- ⑥ Customer can start using the system after every spiral.
- ⑦ Each spiral consists of a Waterfall model.

* Advantages :-

- Useful for large projects
- Customer requirements evolve over a period
- Early available of usable system.

* Disadvantages :-

- Total blue print should be available in the beginning
- Basic concepts in the underlying layer can NOT be changed.

Q.No.

Q.No.

Q.No.

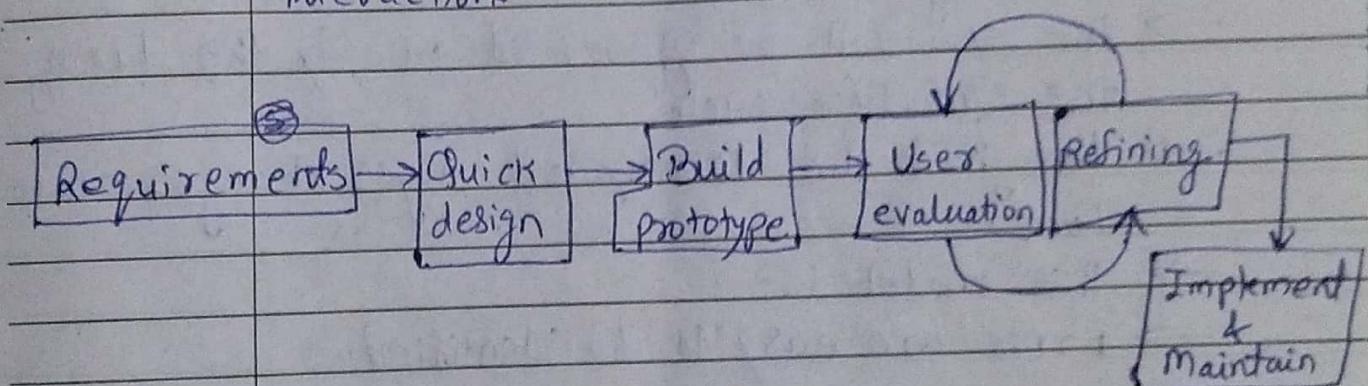
What is prototype Model?

→ A prototype model is a representation of a real life situation, which can be evaluated by the user.

② A prototype is developed based on the initial understanding of the customer requirements.

③ A visible working prototype helps customer to define the requirements.

④ Prototype model is used for eliciting customer requirements as well as identifying user-machine interactions.



* Advantages :-

- ① can be used when customer is not sure about what he wants
- faster way of finalising the requirements
- Useful for new technologies & domains

* Disadvantages :-

- A prototype if used in a production environment, may lack quality or maintainability.

Q.No.

Q.No.

Q.No.



What is Incremental model?

- ① It is a process of SDLC where requirements are broken into multiple standalone of software development life cycle.
- ② It can be done in steps from analysis, design, implementation, testing, maintenance.
- ③ Highest priority requirement is tackled first
- ④ Once the requirement is developed, requirement for that increment are frozen.

* Advantages :-

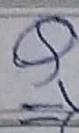
- The s/w will be generated quickly throughout the s/w life cycle.
- Throughout the development stages changes can be done.
- This model is
- Errors are easy to be identified.

* Disadvantages :-

- Rectifying a problem in one unit requires correction in all the units and consumes a lot of time.
- Each iteration phase is rigid & does not overlap each other.

Q.No.						Q.No.			
-------	--	--	--	--	--	-------	--	--	--

प्र.क्र.
Q.No.



What is Agile S/w development?

① Agile S/w development methodology is one of the simplest & effective processes to turn a vision for a business need into Software Solution.

② Agile Features :-

- Highest Priority - Customer satisfaction
- Welcome changing requirements - Even late in the development.
- Deliver working software frequently.
- Business people & developers work together.
- Build projects around motivated individuals.
- Working software is the primary measure of progress.
- Simplicity is essential.
- Best performance of any individual will emerge from self-organizing teams.

* Agile SDLC Models

① SCRUM

→ Sprints

→ Daily Stand-up meetings (^{Progress}
^{problems & plan})

→ SCRUM Master

→ Backlog Sprints

Note:- Type of requirements → Stated
→ legal

Q.No.											
प्र.क्र. Q.No.											

प्र.क्र.
Q.No.

Day 84

Software Testing

Q What is S/w testing?

⇒ Process of identifying Quality and Accuracy of the Software under Test.

* Testing Objective

→ Uncover as many bugs as possible

→ Demonstrate that software matches it's requirement specification

→ Validate the quality of software with minimum cost and effort.

* Principles of S/w testing

→ Testing indicates the present defects

→ Early Testing is NOT possible.

→ Exhaustive testing is NOT possible

→ Defect clustering

→ Pesticide Paradox

→ Testing is context dependent

→ Absence of errors fallacy.

* Specified names :-

① How to test ← Methods

② When to test ← Levels

③ What to test ← types

Q.No.					Q.No.			
-------	--	--	--	--	-------	--	--	--

* Types of SW testing?

- ① Unit testing.
- ② Integration testing.
- ③ System testing.
- ④ User Acceptance testing.

① Unit testing.

→ checks if SW components are fulfilling functionalities or not

② Integration testing

→ checks the data flow from one module to other module.

③ System testing

→ evaluates both functional & non-functional needs for the testing

④ Acceptance testing

→ checks the requirements of a specification or contract are met as per its delivery.

Q) Why to test early?

→ This helps to capture & eliminate defects in the early stages of SDLC.

Q.No.



Types of System Testing

① Usability Testing

→ mainly focuses on the user's ease to use the application, flexibility in handling controls & ability of the system to meet its objective.

② Load testing

→ It is necessary to know that a s/w solution will perform under real-life loads.

③ Regression testing

→ It involves testing done to make sure none of the changes made over the course of the development process have caused new bugs.

④ Functional testing

→ It involves is a type of s/w testing which is used to verify the functionality of the s/w application.

⑤ Recovery testing

⑥ Security testing

⑦ Performance testing

Q.No.

Q.No.

प्र.क्र.
Q.No.

Type of user Acceptance testing.

- ⇒ Alpha & beta testing
- ⇒ Contract Acceptance testing
- ⇒ Regulation Acceptance testing
- ⇒ operational Acceptance testing

* Methods of s/w testing

- ⇒ static testing
- ⇒ Black box testing
- ⇒ White box testing
- ⇒ Gray box testing
- ⇒ Ad-hoc testing
- ⇒ manual testing
- ⇒ Automated testing

Q.No.							Q.No.				
-------	--	--	--	--	--	--	-------	--	--	--	--

प्र.क्र.
Q.No.

Day 5

- * S/w testing - Documentation
- ⇒ Test plan
- ⇒ Test Scenarios
- ⇒ Test cases
- ⇒ Traceability Matrix

Q Verification Vs Validation?

⇒ ① Verification :-

Verification in S/w testing is a process of checking Software requirement specification (SRS) in order to check if the S/w is build according to the requirements. It is Quality Assurance (QA) - reviews.

② Validation :-

Validation in S/w testing/Engineering to make sure that S/w meets the Business requirements.

Validation is QC (Quality Control) - testing

Verification



Am I building
the product right

Validation



Am I building the
right product?

incurring → गुणीता के अनियन्त्रित परिवर्तन

Q.No.

Q.No.

R.S.
Q.No.

Q QA Vs QC Vs testing

① QA (Quality Assurance)

- Proactive (before writing code)
- Process Oriented (how to perform activity)
- How to perform an activity
- Ensure that people are following process
- Training
- Audits (After writing code).
- Select & promote the use of right tools.

② QC (Quality Control)

- Reactive
- Detect bugs
- reviews, walkthroughs & Inspection

③ Testing

- subset of QC.

Q Cost of Quality

Total Cost of
Quality

Cost of
Good Quality

Cost of
Poor Quality

Prevention

[include cost
of training
developers on
writing secure
& easily
Maitainable
code]

Appraisal

[cost of creating
test cases, setting
up testing
environments,
revisiting testing
requirements]

Internal
failure

[cost incurred
in fixing
detected just
before
delivery]

External
failure

[product
support cost
incurred by
delivering
poor quality
Software]

Q.No.							Q.No.				
-------	--	--	--	--	--	--	-------	--	--	--	--

प्र.क.
Q.No.



Software development Test Life cycle (STLC).

Phases \Rightarrow ① Software Analysis

② Test planning

③ Test case development

④ Test Environment setup

⑤ Test Execution

⑥ Test closure.

* V - model.

Requirement
Analysis

Functional
Specification

High level
design

Detailed
Design

Code

User Acceptance
Testing.

System
Testing

Integration
Testing

Unit test

Q.No.

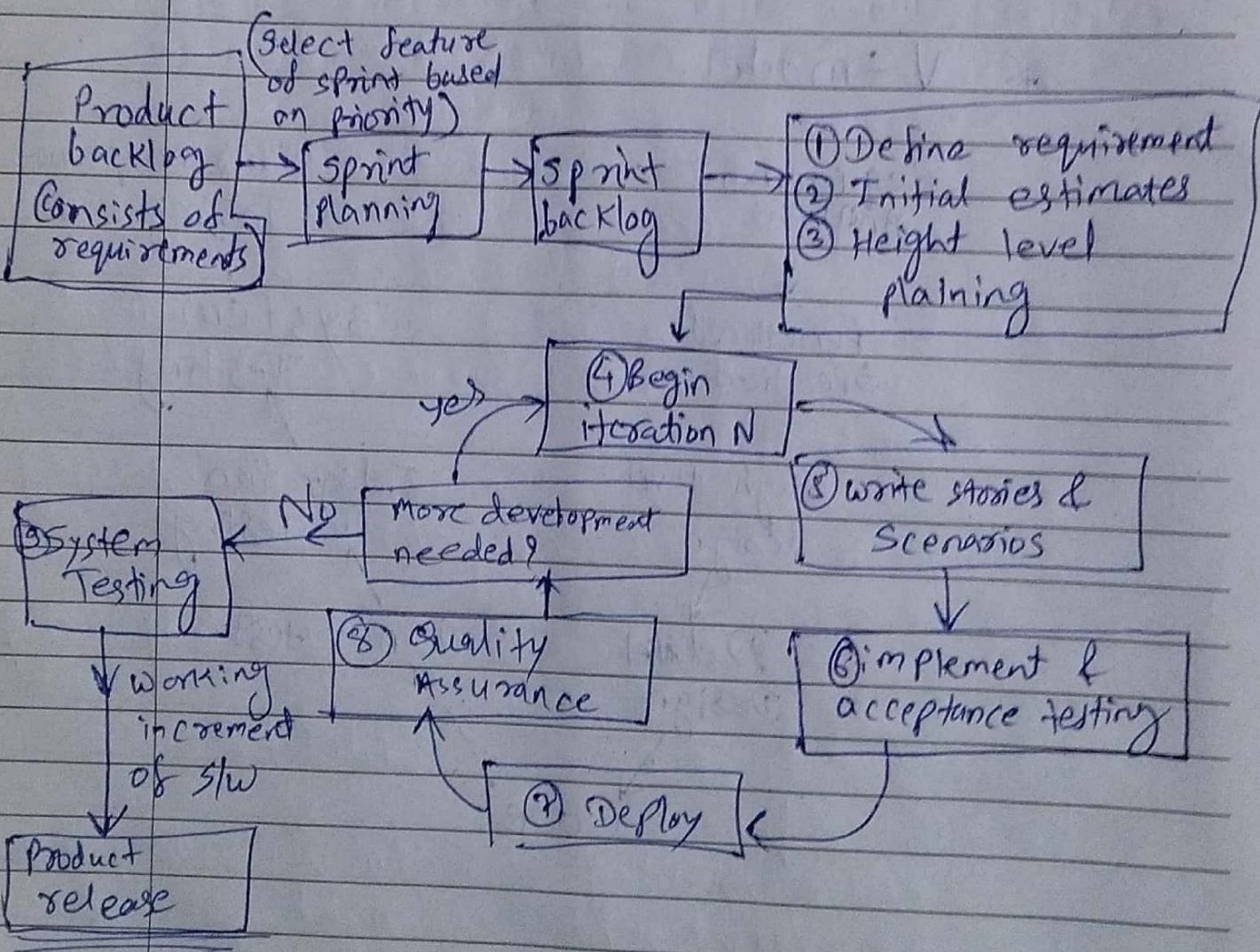
Q.No.

प्र.क्र.
Q.No.Day 3

- * There are various agile methods:

(1) Scrum

\Rightarrow SCRUM is a agile development method which concentrates specifically on how to manage tasks within a team-based development environment.



Q.No.						Q.No.			
-------	--	--	--	--	--	-------	--	--	--

प्र.क्र.
Q.No.

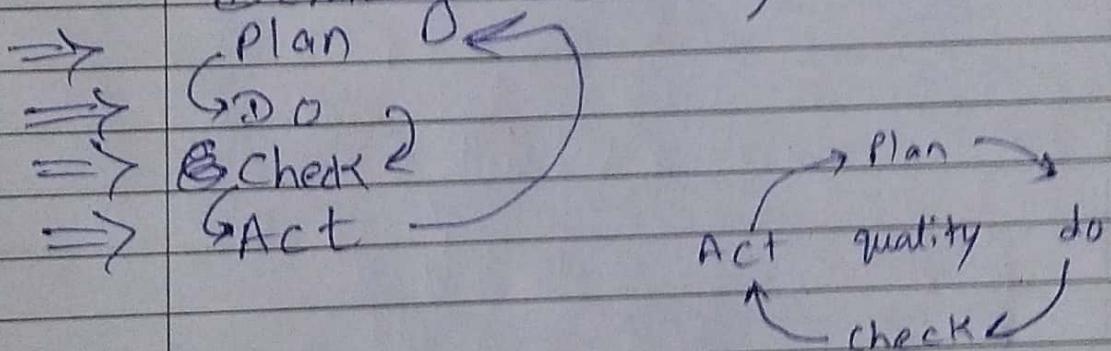
(2)

Extreme programming

(not defined)

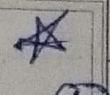
- ⇒ ① OO is the preferred development paradigm.
- ② only 4 phases - planning, design, code & Test
- ③ Using User stories that describe the functionality.
- ④ Design to follow "keep it simple" approach.
- ⑤ Coding - continuous 'refactoring' & 'pair programming'.

* Deming PDCA cycle of Quality



Q.No.												Q.No.								
-------	--	--	--	--	--	--	--	--	--	--	--	-------	--	--	--	--	--	--	--	--

प्र.क्त.
Q.No.



Analysis & design Models

- ① SSAD (structured system Analysis & design method)
- ② OOAD (object oriented analysis & design).

* Tools used in SSAD

- ① ER-Diagram

* OOAD

- ① More connected to real world examples.
- ② Systems are thought about as objects & their interactions.
- ③ Object belong to classes.