

Wednesday

classmate

Date _____
Page _____Unit-1 Introduction to System Engineering.UML - Unified Model language

How to develop a system, methodology, structure of Softwares.

to develop quality product
to develop software

Prototype → synonyms or similar model of final product.

IEEE Definition.

→ "Software engineering is the application of Systematic, disciplined, quantifiable approach to the development, operation, and maintenance of Software; that is, the application of engineering to Software."

Pepsi → taste
Requirement

Software Crisis

- It is often the case the Software product.
 - fail to meet user requirements.
 - Expensive Cost
 - Difficult to alter, debug, and enhance. MM[change]
→ not able identify errors
 - Often delivered late.
 - Use resources non-optimally.

Software Crisis (cont.)

Laptop or Desktop =

Rs 45,000/-

Rational suite node locked =

Rs 3,14,600/-

Rational suite floating license

Rs 6,03,200/-

Hw cost ↑

Sw cost

8/02/24

Rom/ROM.
Processor.OS
Obj-Oriented
Structured system
hardware
software
embedded software↓
Use for some particular task only.

1960

Year →

2018

Relative Cost of Hardware & Software

maturity

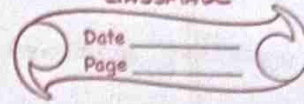
CMM - Capability Maturity Model

28% successful

23% cancelled

49% Delayed or cost overrun

devops = development + operations
classmate



Factors Contributing to the Software Crisis

- larger problems
- poor project management
- lack of adequate training in software engineering.
- increasing skill shortage.
- Low productivity improvements

legacy software

system software

embedded software

firmware software (Air conditioner remote which

application software handel from anywhere)

web^{app} software

desktop applⁿ

Exploratory style

bugs

10/02/24

Set of instructions to acquire some input to produce some desired output by processing → software

- * set of documents, such as the software manual, meant for users to understand the software system.

2 Types of software

- (i) Generic software → which can be used by any requirement are very common, one fairly stable

classmate

Date _____

Page _____

(2) Customized Software - For particular users, organization, ex- ISRO
Requirements are being unique to customers

attributes of
good →
students

Good Software

Software has number of attributes which decide whether it is a good or bad.

Teacher
Punishment
to student
behaviour
→

defn of a good software changes with the person who evaluates it.

Software is required by the customer, used by the end users & of an orgaⁿ & developed by SE

→ Each one evaluate the diffⁿ attributes

* good attributes

(1) Maintainability - Software must evolve to meet changing need

Adhar
dependable (2)
if not working

Dependability - must be trustworthy

(3) Efficiency - sf should not make wasteful use of right time per result mile System resources [output on time]

(4) Usability - S.f must be usable by the users for which it was designed

engineering - A Appln of Science to the soln of practical

Software engine = The appln of CS to Building Practical Software system

How we can develop the Software/Appln

Characteristics

- * dual role - product but also a vehicle for delivering a product.
- (medical software)
- * is logical rather than a physical system element.
- * differ considerably from those of hardware.
- * Software doesn't "wear out".
- * Most S.F is custom-built

* changing nature of software

- (1) System Software - A colleⁿ of program written to service other progr at system level
- (2) Real time - monitor real world events as they occur
- (3) Business
- (4) Engineering & scientific - store test are analyzing by AI and processing by scientists for output
- (5) Embedded Software - ROM & used to control product
- (6) Artificial Intelligence (AI) software - games, stymee, pattern recogn
- (7) Internet Software.

programming.

- Individual writes program
- One person, one computer
- Well-defined Problem
- programming in the small

Software Engineering

- Individual writes program Components
- Team Assemble complete code.

classmate

Date _____

Page _____

layered - pressman's View technology

- tools
- methods
- process model
- a "quality" focus

Stages

Software Development Methodologist

- | | |
|---------|-------------------------|
| Stage 1 | (1) Planning |
| 2 | Analysis |
| 3 | Design |
| 4 | Implementation |
| 5 | Testing and Integration |
| 6 | Maintenance |

15/02/24
Thursday

Scientific method to develop a Software
At what extend you provide
a function = scope. ^{classmate} ^{quality} ^{Product}
Date _____
Page _____
2 - Intro to Software Engineering

* We can modify linux library.

* System Building Blocks and Interfaces. (a)

- Program risks
- Technological performance limits
- Interfacing requirements
- To make trade off analyses among design alternatives.
- hierarchy of modern system

* Concept of hierarchic Systems

* System Levels Structure :-

System → Subsystems → Components → Subcomponents
→ Parts

1) System → hotspot, fuelling station, CNG, Communication line, power grid. (Electricity).
System is set up from where things are distributed.

ex- Computer (Input, process, output) ^{keyboard} CPU, mouse
CPU, passenger aircraft all are Systems.

classmate

Date _____
Page _____

3) Components → Requirement documentation
middle level of system

(4) Subcomponents

(5) PARTS (lowest level)

* Building blocks

- means for classifying system constituents according to
 - functional characteristics (user ke functionality chahiye)
 - physical characteristics (ATM card device by account holder is there which we fetch).
- Useful for visualizing potential architectures of System Concepts. (report part).

21/02/24
Wednesday



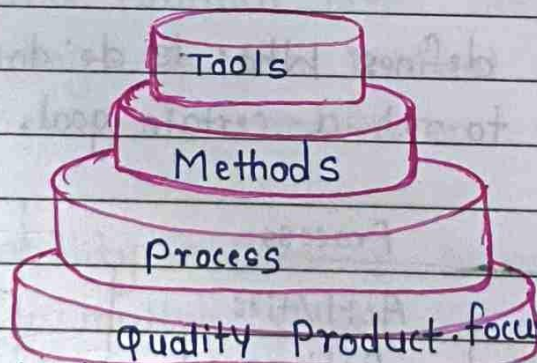
Nature of Software

- (1) Software takes on a dual role, it is a product and at the same time, it is a vehicle for delivery of product.

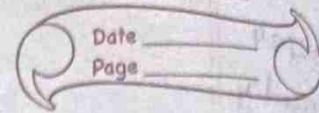
Characteristics of Software

- (1) Software is logical than physical system.
(2) Software is developed or engineered, it is not manufacture.
(3) Software doesn't ~~not~~ wearout.
(4) A software is composed of several reusable components.

Software engineering as layered technology



- * Product that is to be made must be a quality product meeting with a users requirement.
- * Process that binds or defines how to build tools for a software and methods available to provide a quality product.



act as a medium between upper two layer and lower layer.

* Defines a framework with a set of activities

(3) → The methods will be particular methods implemented to develop & test and softwares

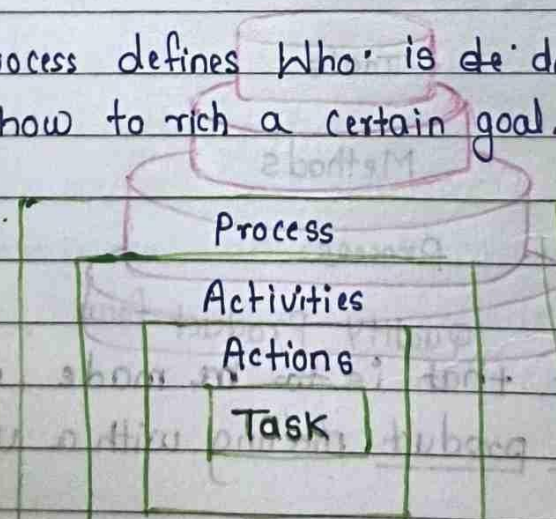
→ Tools are automated softwares that minimizes development, complexity, duration

eg- android studio, vs code, netbeans, microsoft visual studio.

* Software process

→ A process is a collection of activities, actions and tasks that are performed when some work product is to be created.

→ A process defines who is doing what, when and how to reach a certain goal.



→ A process framework establishes the foundation for computer software engineering process by identifying a small number of activities that are appropriate for all software projects irrespective of their size & complexity.

* Key principles of software engineering.

1) Modularity - Breaking systems into smaller parts to understand requirements better and to minimize the complexity of the system.

2) Abstraction - Hiding the implementation of a component and exposing only the necessary functionality to other parts of the software.

3) Encapsulation - Wrapping of the data and function into single unit and protecting it with internal state of an object from external modification.

4) Reusability - Creating components that can be used in multiple projects that can be used in multiple projects which can save time & resources.

regularly
5) Maintenance - Updating and improving software to fix bugs, to add new features, security, vulnerability.

Single sign on

classmate

Date _____

Page _____

- e) Continuous Integration & deployment - Continuously integrating the code changes and deploying them into production environment.

Wednesday - 6/03/24

* Software Quality - can be defined as 'conformance to requirements' and/or 'fitness of use'. Quality is based upon the customer's actual experience with the product or service, measured against his or her requirements.

- Stated or unstated
- Conscious or merely sensed
- technically operational
- entirely subjective

McCall's Quality Model

Perspectives are :-

↑ maintainability, Testability, flexibility.

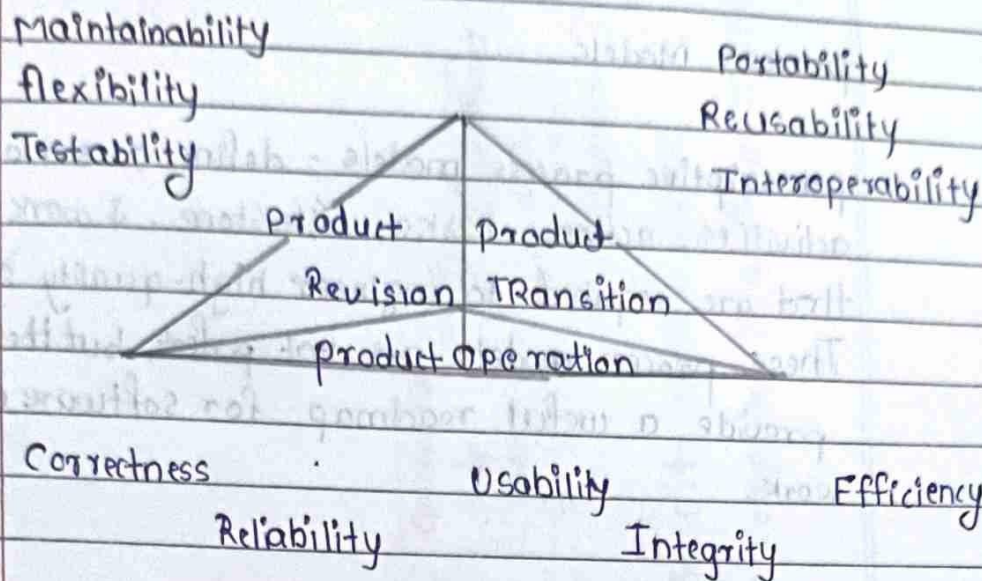
- Portability (1) Product revision (ability to change).
 Reusability (2) Product transition (adaptability to new environment)
 Interoperability (3) Product operation (Basic operational characteristics)

↓
 Correctness, reliability, efficiency, Integrity, usability.

9001 → develop. service is providing

classmate

Date _____
Page _____



7/03/24

Thursday

Berry Bohems quality Model

Perspectivie of quality measure.

- (1) AS-Is - Utility (functionality of the quality model).
- (2) Maintainability.
- (3) Portability.

Q1) ISO 9001 - design, development, service

(one professional group)

→ * ISO [International Standard Organization] is a Consortium of 63 countries

- * Established to formulate and foster standardization
- ISO published its 9000 Series of Standards in 1987

* ISO 900 Series: ISO 9001, ISO 9002 and

Now ISO 9003.

ISO 10000 Series

↓ demolished in 2004. (it was not appropriate).

3/04/24

Wednesday

classmate

Date _____
Page _____User experience
UX

process Models

prescriptive process models - define a set of activities, actions, tasks, milestone, & work products that are required to engineer high-quality software. These process models are not perfect, but they do provide a useful roadmap for software engineering work.

The WATERFALL MODEL :-

The waterfall model, sometimes called the classic life cycle, suggests a systematic sequential approach to software development that begins with customer specification of requirements & progresses through planning, modeling, construction & deployment

SDLC Software development life cycle

(i) The waterfall Model

not feedback activity
It is not practical.

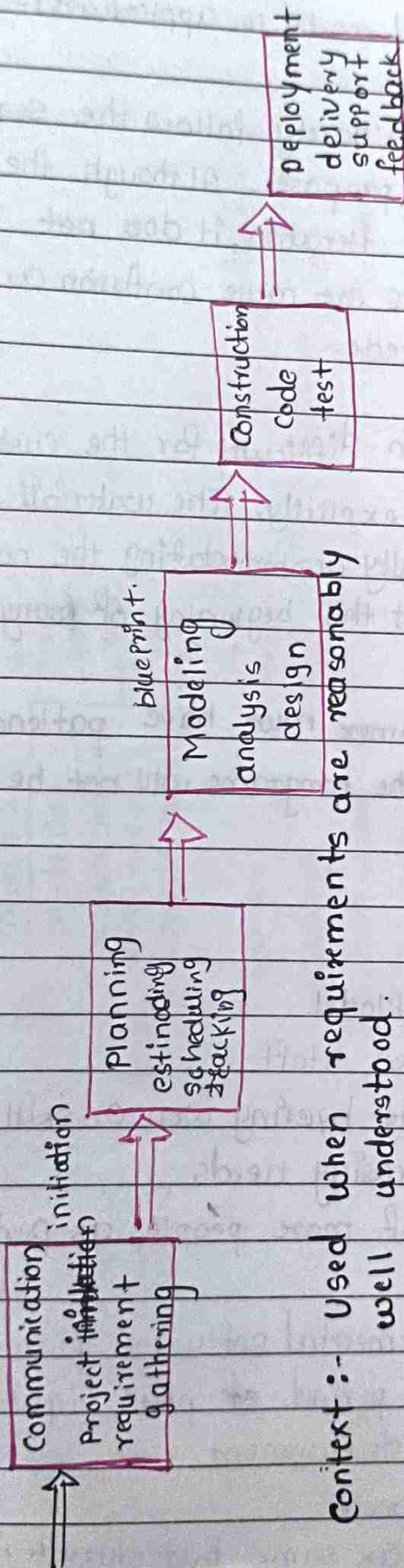
Use for small system

Smaller instructions are their & fixed

classmate

Date

Page



Context :- Used when requirements are reasonably well understood.

Advantages - It can serve as a useful process model in situations where requirements are fixed & work is to proceed to complete in a linear manner.

* The problems that are sometimes encountered when the waterfall model is applied are:-

medical
billing
fail to
explain.

(1) Real projects rarely follow the sequential flow that the model proposes. Although the linear model can accommodate iteration, it does not so indirectly. As a result, changes can cause confusion as the project team proceeds.

(2) It is often difficult for the customer to state all requirements explicitly. The waterfall model requires this & has difficulty accommodating the natural uncertainty that exist at the beginning of many projects.

RAD
Rapid
Appln
Development

(3) The customer must have patience. A working version of the programs will not be available.

RAD Model

* INCREMENT Model.

Starting less staff
adding or high hiring well or skill according to
increasing needs.
increment of more people as per requirement.

RAD = incremental software process

Short period of phase required for
development.

⇒ high speed.

⇒ phases are same but duration is less.

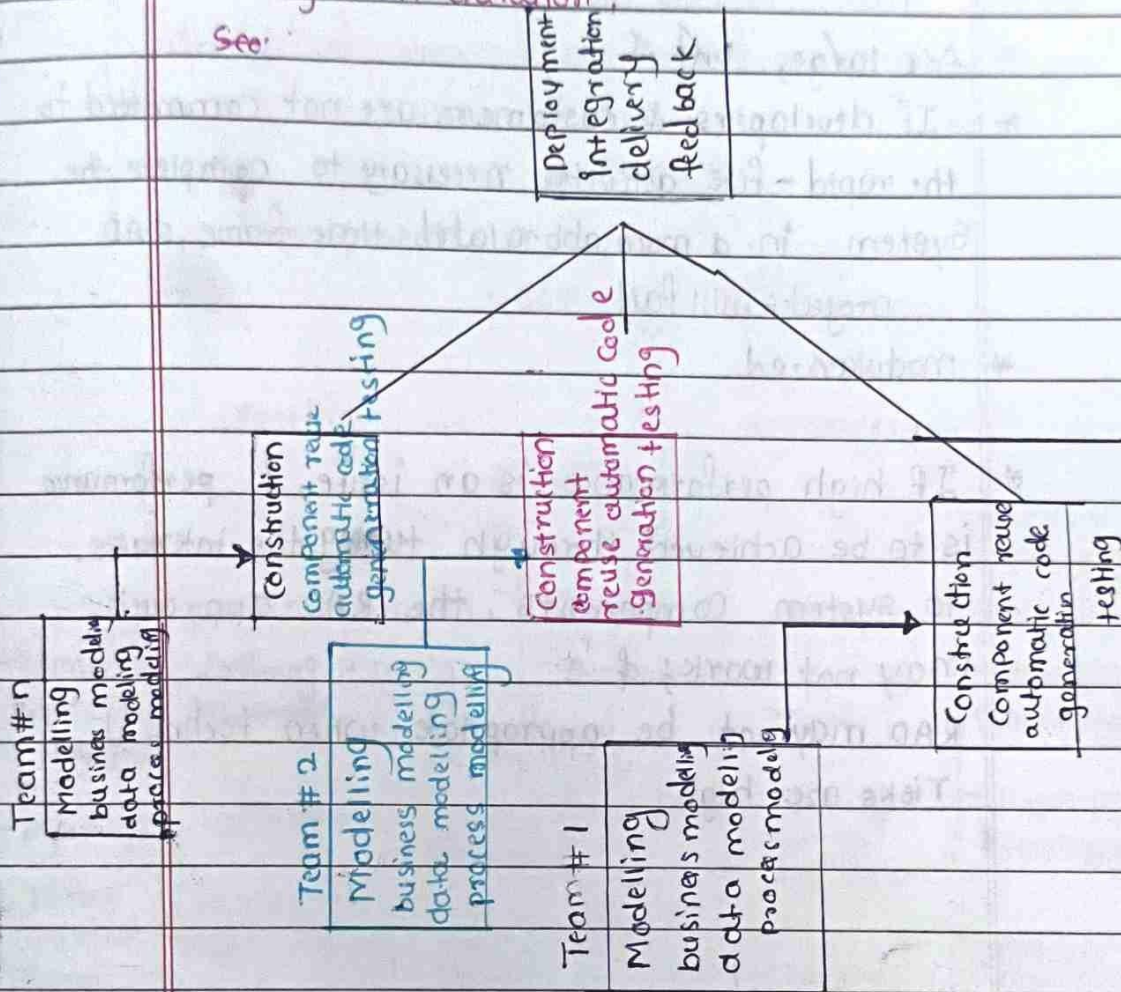
debug
ideallyscope is limited
process are defined.

classmate

Date _____
Page _____

Context → JP requirements are well understood & project scope is constrained, the RAD process enables a development team to create a "fully functional system" within a very short duration.

See:



Friday
5/04/24

Projects are divided into
different group

classmate

Date

Page

drawback.

- * sufficient no. of people are include, or sufficient human resources to create the right no. of RAD team
- * for large, but is
- * If developers & customers are not committed to the rapid-fire activities necessary to complete the system in a much abbreviated time frame, RAD projects will fail
- * modularized.
- * If high performance is an issue, & performance is to be achieved through tuning the interfaces to system components, the RAD approach may not work; &
- * RAD may not be appropriate when technical risks are high.

Wednesday

10/04/24

classmate

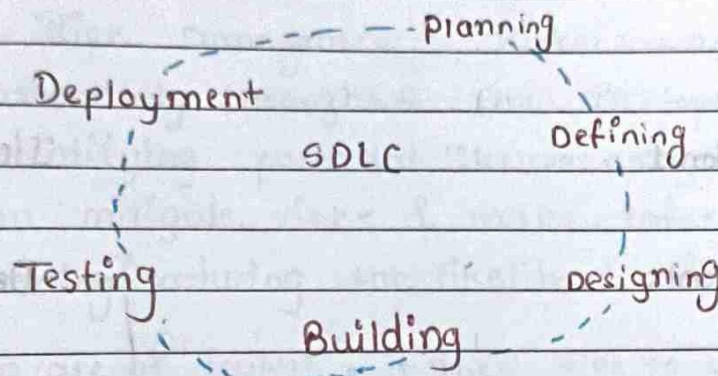
Date _____

Page _____

* Software Development life cycle

→ Systematic process

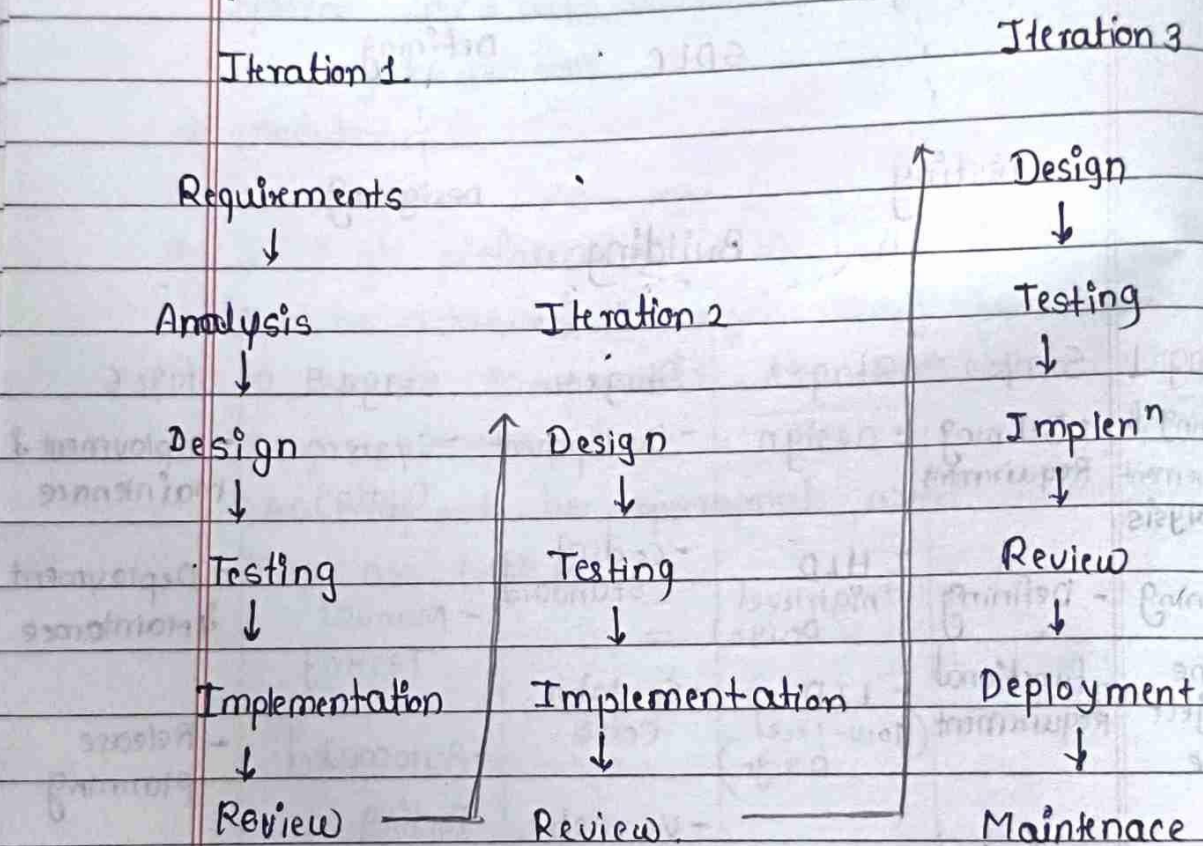
→ SDLC is a structured process that is used to design, develop, & test good-quality software



Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6
- Planning & Requirement Analysis	- Defining Requirements	- Design	- Development	- System Testing	- Deployment & Maintenance
- Planning	- Defining	- HLD (high level Design)	- coding Standard	- Manual Testing	- Deployment & Maintenance
- Define project Scope	- Functional Requirement	- LLD (low-level Design)	- Scalable Code	- Automated Testing	- Release planning
- Set objectives & Goals	- Technical Requirement		- Version control		- Deployment Automation
- Resource Planning	- Requirement Reviews & Approval		- code Review		- Maintenance
					- Feed back

2) Iterative SDLC Models

In software development, choosing the right SDLC model is crucial for success. Among the various approaches, the iterative SDLC model stands out as a flexible & efficient methodology that promotes



forgot functionality can be added in another iteration.

Key principles

- (1) Increment progress - The iterative model emphasizes incremental development, breaking down the project into manageable parts.

classmate

Date _____

Page _____

evolution

(2) Flexibility & Adaptability :- one of the model's strength is its adaptability to changing requirements. Developer can easily incorporate.

forgot
password
change
can
be
included

(3) Continuous Evolution

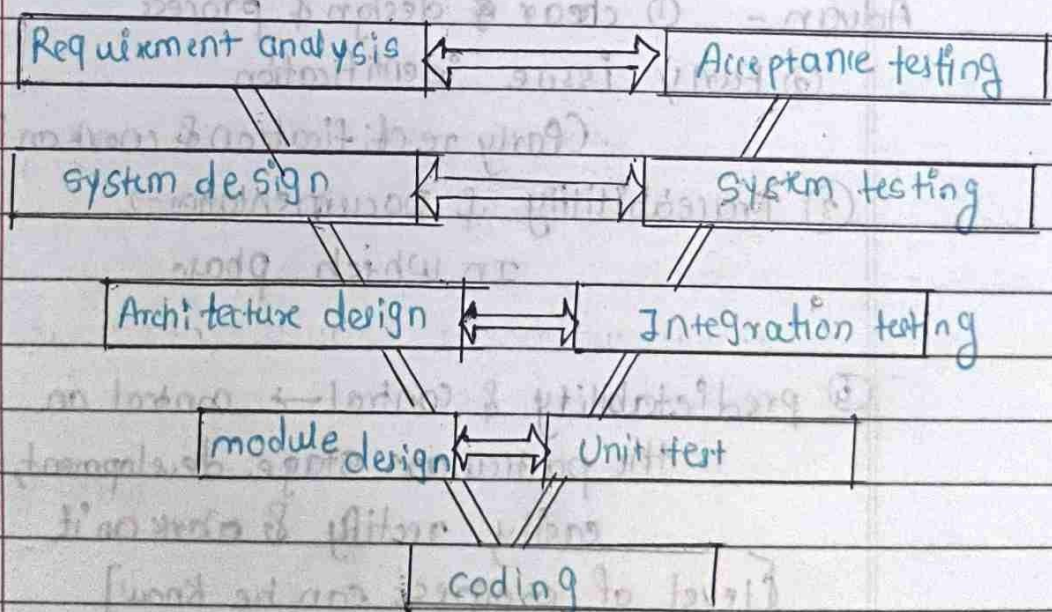
(4) Risk Management - Risks are addressed proactively throughout the development process. By identifying potential issues early on, the team can mitigate risks & make informed decisions, reducing the likelihood of project setbacks.

In case of failure, it gives tips to utilize the problem.

Thursday
18/04/24

Parallel Development Testing

V-models (Verification & Validation Models) in SDLC



- ① White Box testing
- ② Black Box testing

Key principles of V-models

- ① Parallel Development & Testing

↓

early feedback is coming out
So we rectify easily & early.

- ② Verification & Validation -

- ③ Traceability → could be a check at which stage the development process stand on, development, Traceable.

Advantages - ① clear & design & process.

- ② Early issue identification

(Early rectification & work on it)

- ③ Traceability & Documentation → In which phase

- ④ predictability & control → control on the particular stage development, easily rectify & work on it [level of progress can be know]

Senanim.
Testing

classmate

Date _____

Page _____

Best practice of V-model.

Bidirectional model (iterative or continuous manner)

- ① Thorough Requirement
- ② effective communication (clear and
Consistent communication betⁿ development &
testing teams is crucial.
- ③ Automated - Testing - leveraging automated
testing tools enhances the efficiency & effectiveness
of the testing process in the V-models.
Seamless execution, repetition.
- ④ Iterative feedback loop

24/04/24

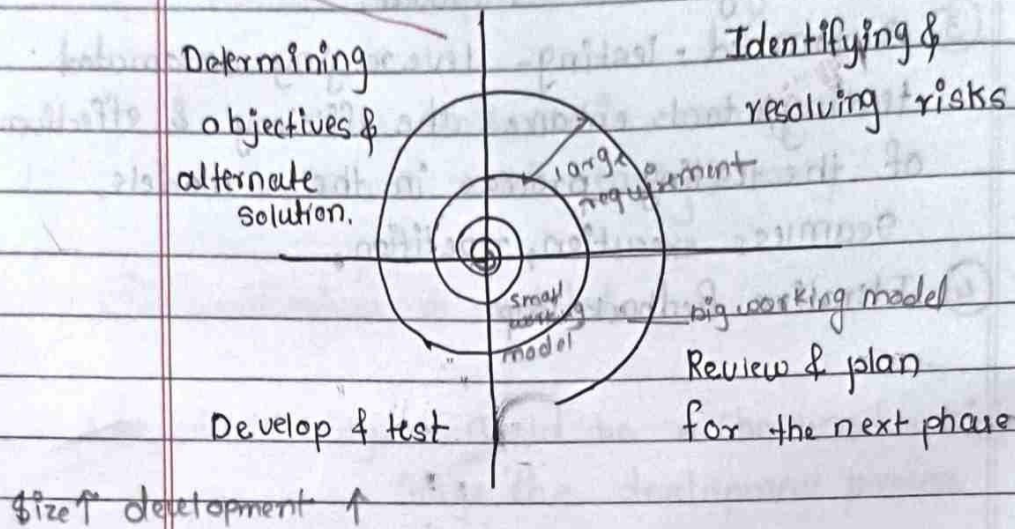
Best practices

- Prioritization & planning
- Continuous integration & testing
- Retrospectives for continuous improvement

classmate

5x Spiral SDLC Model

- combines the idea of iterative development with systematic aspects of the waterfall model.



① identify the requirements

Key principles of spiral

- (1) Iterative development - The spiral model embraces iterative development, allowing for the incremental release of the product. each iteration, or spiral, includes planning, risk analysis, engineering, testing & evaluation process.

Power failure.

going another

People risk

web app

Linux

OS

budget

risk

india

Risk

- (2) Risk Management :- fundamental component, each iteration begins with a risk assignment & the project progresses based on addressing skilled labour, control team can better high-priority risk

Today's
malware
Trojan
NOTES

edits Rina
classmate

Date _____
Page _____

Troy

IT COPIER

(3) flexibility :- highly adaptable to changes in project/software

(a) Continuous Evaluation - Evaln is integrated into every phase of the Spiral Models, After each iteration, the project is reviewed to assess progress, identify potential risks, & determine the next steps.

Advant -

- (1) Risk Mitigation - utilizing the failure, managing
- (2) flexibility in Requirements
- (3) High-Quality product.
- (4) Client Involvement
- (5) Regular review Meetings
- (6) collaborative Team Communication.
- (*) cross functional Teams

Improve
collaborability.

6) Agile Model.

```

graph TD
    Requirements --> Design
    Design --> Develop
    Develop --> Test
    Test --> Deploy
    Deploy --> Review
    Review --> Requirements
  
```

Collaborative Communication Between ^{classmate} team.

Date _____
Page _____

It is continuous process
not a specific methodology but rather a set of principles
& values outlined in the Agile Manifesto.

→ Agile = allow to change / flexible to mold in
environment / changeable.

The Agile manifesto prioritizes individuals &
interactions, working solutions, customer
collaboration, & responding to change over
rigid processes & documentation.

25/04/24
Thursday.

classmate
Date _____
Page _____

Chapter No. 4

Requirements Engineering Process in Software Engineering.

→ Don't have ~~an~~ large expectation

Requirement engineering - is the process of identifying, eliciting, analyzing, specifying, validating, & managing the needs and expectations of stakeholders for a software system.

R.E process

Feasibility Study

Requirements Elicitation

Requirements Specification

Requirements Verification & Validation

Requirements management

1x Feasibility Study

The feasibility study mainly concentrate on below five mentioned areas below. Among these Economic feasibility study is the most important of the feasibility analysis & the legal feasibility study is less considered feasibility analysis.

21/05/24

classmate

Date _____
Page _____

- = economical feasibility
- = operational feasibility
- = technical feasibility

* Requirements Elicitation.

Interview

Survey

Focus Group

Observation - observing stakeholders in their work environment to gather information about needs & expectations.

Prototyping - The technique involves creating a working model of the software system, which can be used to gather feedback from stakeholders & to validate requirement.

* Requirement Specification.

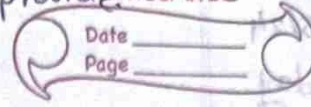
includes ER diagrams, data flow diagrams (DFDs), function decomposition diagrams (FDDs), data dictionaries, etc.

ask
what system should do to us.

1) Functional - these describe what the software system should do. They specify the functional that the system must provide such as input validation, data storage, & user interface.

2) Non-functional Requirement - These describe how well the software system should do it. They specify the quality attributes of the

what system should provide
how system should provide



system, such as performance, reliability, usability, and security.

(3) constraints :-

(4) Acceptance Criteria :-

4) Requirements verification & validation

(1) Verification - It refers to the set of tasks that ensure that the software correctly implements a specific function.

(2) Validation - It refers to a different set of tasks that ensure

① The requirements should be consistent with all the other requirements i.e. no two requirements should conflict with each other.

② practically achievable.

③ complete in every sense.

3/05/24
Friday

SRS - Software Requirement Specification

Android versions 1. (add name version).

5) Requirements Management

order
software
books

process of analyzing, documenting, tracking, prioritizing, and agreeing on the requirement & controlling the communication with relevant stakeholders.

1) Tracking & controlling changes

change are been controlled where we have break

2) Version control - Version of documents

3) Traceability - we can check whether it is complete

4) communication -

5) Monitoring

Notes

Provided

By

Vaibhavi Jadhav

©NotesSociety