SOFTWARE ENGINEERING

Date...15/2/25...

Software:- Set of instructions, data or program used to operate computers & execute specific tasks.

Information System:- They are integrated user machine system that provides right information to the right person for planning organising, coordinating & controlling & decision making within organisation.

* CBIS → Computer Based Information System
* ICT → Information Communication Technology

Types of Software:

Documents
- Problem Introduction (Overview)
- Problem Solution in the form of algorithm.
- Word Breakdown Structure of problem
- DFD (Data Flow Diagram) → level 0 / Context Diagm. → level 1 → level 2
- Entity Relationship Diagram
- State Transition Diagram
- UML Cases (Unified Modeling Language)
- Input Screens / Output Screens
- Test Cases
- References
- Glossary

Q. Explain Software engineering is a document driven science.

Aparts from this, following manuals are required -

1) System Manual

Spiral

Date. 17/2/22

2) Command Manual
3) User Manual (training purpose of end users) for smooth operation & maintainance of the software system)

Software Classification based on Distribution

1) Retailware (FMCG)
2) Original Equipment Manufacturing Software
3) Freeware
4) Open Source Software
5) Shareware
6) Demoware
7) Spyware
8) Addware
9) Crippleware

Quality

a) Authentication
b) Authorization
c) Binary

① Dependability
a) correctness
b) Reliable
c) Testable
d) Maintable
e) ROBUST
f) Usable
g) Available
h) Safe

② Performance
a) Memory → Internal / External
b) CPU

③ Security

$$(\alpha) = \frac{MTTF\ (Mean\ Time\ To\ Fail)}{MTTF + MTTR}$$

↳ (Mean Time To Repair)

Date. 22/2/23

BPO → Bussiness Process Outsourcing
KPO → Knowledge Process Outsourcing

* Authentication
  → Login & Password
  → Smart Cards
  → Tokens
  → Biometeric Authentication

* Software Engineering

ISO, ANSI, ROI, IEEE, IBM
IEEE → Institute of Electrical & Electronic Engineers.

5 to 7 yrs

1. Development
2. Operations
3. Maintainance.

Software Engineering is Systematic, displined, quantifiable, approach of software development, operations & maintainance

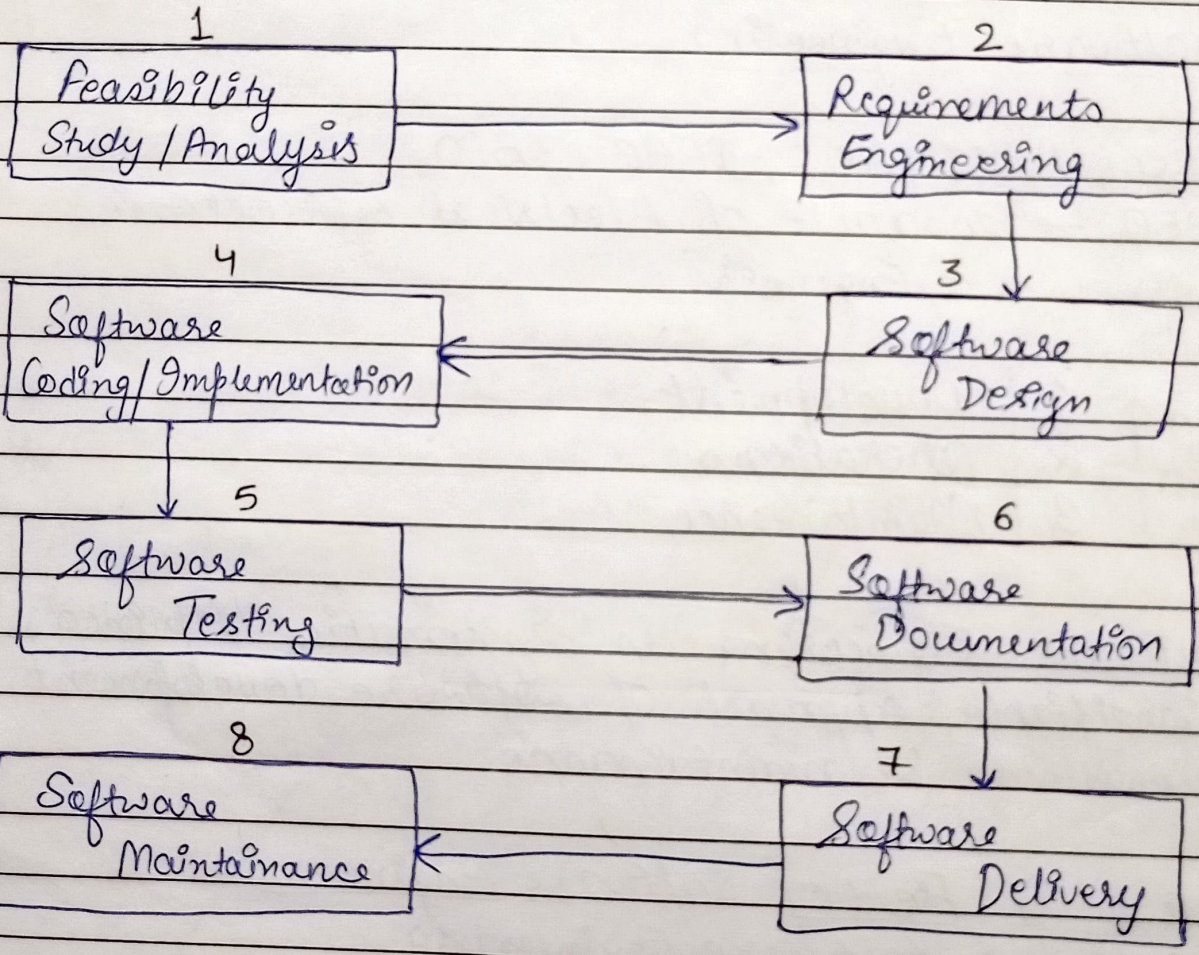
Career In Field of Software Engineer
People in Software Engineering
① End User
② Stakeholder (other owner or having ownership)
③ Sales & Marketing Personel
④ Managers
⑤ Domain Experts
⑥ Analyst
⑦ Software Designers & Architects

Spiral

Date. 23/02/23...

⑧ Technical Technology Experts
⑨ Programmes
⑩ Tester & Detesters
⑪ & User support staff
⑫ Software Engineers

Software Process & Process Models



Mature Process → predictable, manageable process, flexible, practical & measurable, planned oriented for the software development.

Process → Process may be defined as a phase-wise action that is carried out in a

Date.....

non-zero but finite period of time to accomplish a pre-defined or a well-established goal.

Software Process - It is a process of developing software application. All development processes starts with some objective or needs.

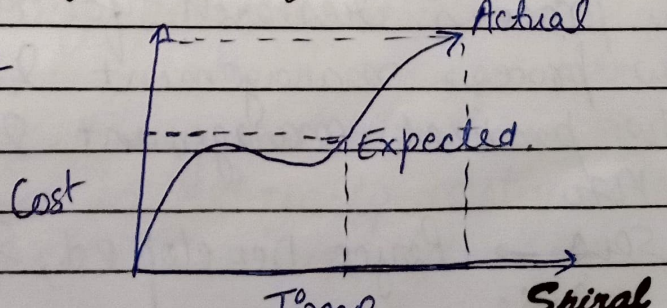
* The development of a solution starts with actions to understand the needs & then proceeds to plan how to meet-out.
* During planning all the available & required process are compared to obtain the idea of feasibility of the solution & the required (feasibility means possibility) resources are then managed.

The software process & its effectiveness may not be important for a short-term organisation but it is a great value for long term object organisation.

Long-term software developers practisioner must pay fair attention to follow formalised & established processes (best practices) for their development otherwise they may compromise on quality of their product & cutting the life span.

Challenges before Software Engineers

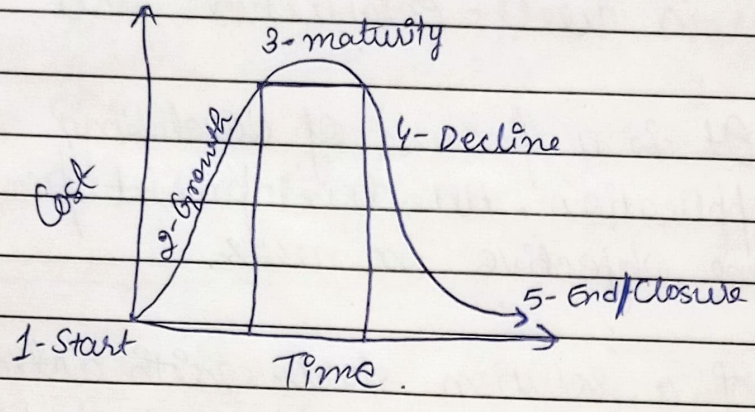
① Rising Cost Project



* BPR ⇒ Business Process Engineering.
Process

Date. 24/02/23...

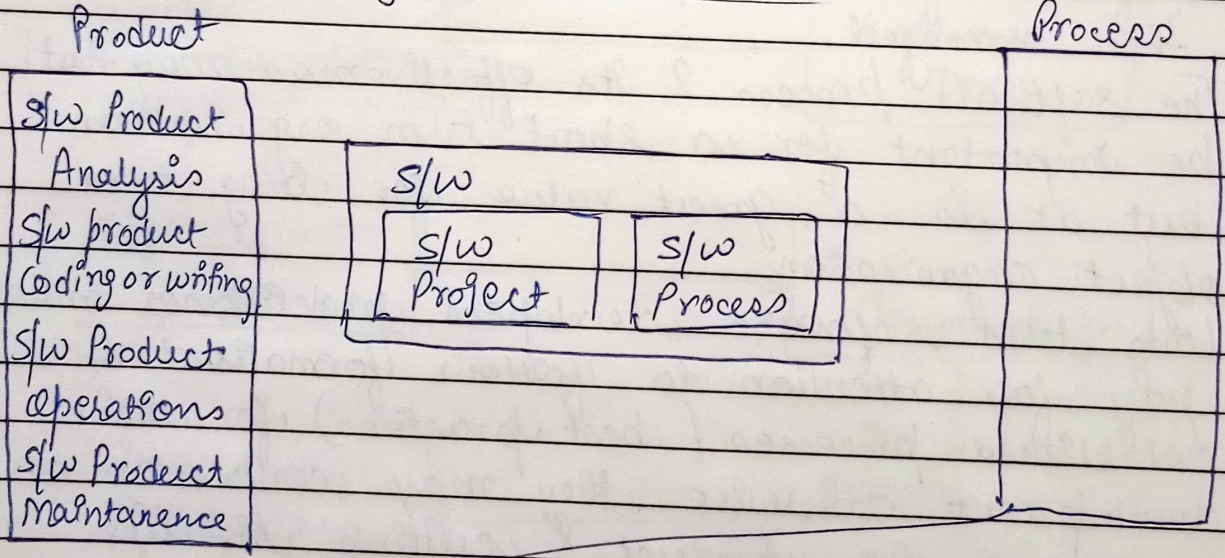
Project life Cycle



② Knowledge of Design Approach -

③ Knowledge of Process Model -

→ Software Engineer ~~Architect~~ Activities -



* S/w Process Development from beginning to the end
* S/w process research for enhancement
* S/w process management & control
* S/w product management & control.

1970

{* SDLC → Royce Developed, Software Development Life Cycle}

Date.....

Models (V-Shaped)

① LSM ③ RAD ⑤ Spiral Model
② PRM ④ INM ⑥ WIN-WIN Spiral Model
⑦ Agile Model ⑧ Iterative Model

* Handling problem at abstraction level
→ Measuring & planning of the project

* Myths / Facts Related to Software -

① S/w easy to change - Myth
S/w very complex to change or modify - Fact

② Computers provides greater reliability than the devices they replaced - Myth.

③ Increasing S/w reliability will ↑ safety - Myth

④ Providing & testing S/w

⑤ Re-using S/w ↑es reliability & safety - Myth

⑥ By eliminating oper. Human errors can be eliminated - myth

Requirement Engineering

Requirement engineering provides bridge b/w domain with its demands & goals & a software & hardware solution. Every mistake in software engineering process is extremely costly. Either in terms of collection or in terms of

Spiral

solution
providing ~~user~~ with a invalid ~~system~~ that reflects your inconsistency.
The main task of requirement engineering is to determine & specify explicitly that ~~is~~ stated & implied need ; i.e. the requirement a system has to fulfil.

Stated Need → functional requirement
Implied Need → Non-functional requirement

These needs have their origin in a certain Context such as stake holders, goals & objectives, assumptions & application domain.

Specification is the task of precisely describing the softwares to be written in a mathematically ~~regress way~~ rigorous way.

The informal knowledge about functionality, behavioural expectation, performance & reliability needs as gathered from the customer & then customer needs to be translated into formal & unambiguous specification that are well-developmented in the SRS document.

Formal → standard format

Requirements Elicitation Techniques & Tools.

1. Observation
2. Interview

Q. Describe interviewing as an art.

Interview

Face-to-face — Telephonic

Unstructured

Structured

(3) Questionare ⇒
- → Open Ended (5-10%)
- → Closed Ended (90-95%)

(4) Delhi Technologies

Q. Describe the role of SRS in the software development process. Also describe the characteristics of a good SRS & describe IEEE 830 standards for drafting SRS.

SRS is used in validation & verification.

Structured Walkthrough
path with landmarks

Specification of the requirement are the informal knowledge about the functionalities, behavioural expectation, performance & reliability needs as gathered from the customer & then such ~~custom~~ cost consumer need should be transeated to formal & ~~unat~~ unambigious specifications that are well-documented in the SRS

"730" SQA → Software Quality
Date.....

SRS document structures to specify product line requirements, a good ~~SRS~~ quality SRS can provide multiple benefits such as establishing the bases for the contractual agreement between the customers & developer & for performing cost, schedule & resource estimates for the developed software project.

SRS is a very important document that supports a base line for verification & validation of a software product.

Characteristics & Qualities of a good SRS.

* Correctness.
* Consistant
* Completeness
* Unambiguous
* ~~Very flexible~~ Varifiable.
* Traceable
* Modifiable / Flexible (Upto certain limit)
* Simple
* Usable
* Understandable
* Relevant with affordability.

1 → Introduction.
- 1.1 → Purpose
- 1.2 → Scope
- 1.3 → Glossary
- 1.4 → References
- 1.5 → Overview

Spiral

Date.. 22/3/23...

Q. Software Engineering is layered Technology.

4 (Tools Layer)
↓
3 (Method/Technique Layer)
↓
2 (Process Layer)
↓
1 (Focus on Quality)

* CASE → Computer Aided Software Engineering Tools.

2- Requirement Specification
2.1 - System Environment (Hardware/Software)
2.2 - User Description
2.3 - User Case Modeling (DFD, E-R Diagrams)
2.4 - Use Case Analysis (STD → state transition diagram)
2.5 - UI Specification
2.6 - Non-Functional Requirements

3- System Evolution.

Software Testing
Search for undiscovered error.

Objective:
1. Immediate Testing
2. Long-Term (Robustness, Reliability, etc)
3. Post-implementation (easy future enhancement)

Spiral

Bug → Error → Fault → Failure.

Date.....

Q. Why documentation is necessary in Software Engineering?

Q. Software development process is document-driven process.

Document = doesn't need reference

Non-Document = needs reference

Ans-2 Phase-I Requirements Engg ⇒ 1. SRSD
2. System acceptance Plan

Phase-II Design ⇒ 1. SDD — ~~System Test Plan~~ S/w Design
2. HLD — Functional Test Plan ~~Module Design~~
3. LLD — Unit Test Plan Program Design

1. S/w Overview
2. Introduction
3. System Manual
4. Command Manual
5. User Manual
6. Test Cases.
7. Delivery

* Process
I) Development
II) Implementation

Delivery / Installation / Implementation

Types

1- Training
2- Conversion
3. Implementation / Installation
a → OVER NIGHT / BING BANG Implementation
→ is best when current implementation has failed
→ is best, when application is very small.

Spiral

Date 22/3/23

Q. Why is a change is always resisted?
* Fear of failure
* Fear of learning
* Fear of unknown
* Fear of position / status
* Fear of power

Software Maintenance (As per 2000 Survey)
Types
I. Corrective (21%)
II. Adaptive (25%)
III. Perfective (50%) / Enhansive
IV. Preventive } (4%)
V. Gold Plating }

Software Maintenance

- Problem / BUG Reporting → Major, Moderate, Minor
- Problem / BUG Analysis
- Patch Creation
- Minor Enhancement
- Minor Adaption
- Minor design alternation
- Version Control
- Corrective Bug Fixing

SCM → Software Configuration Management
↳ defines a boundary for maintenance

Software Development Process Model (SDLC)
Tool of Project Management

Spiral

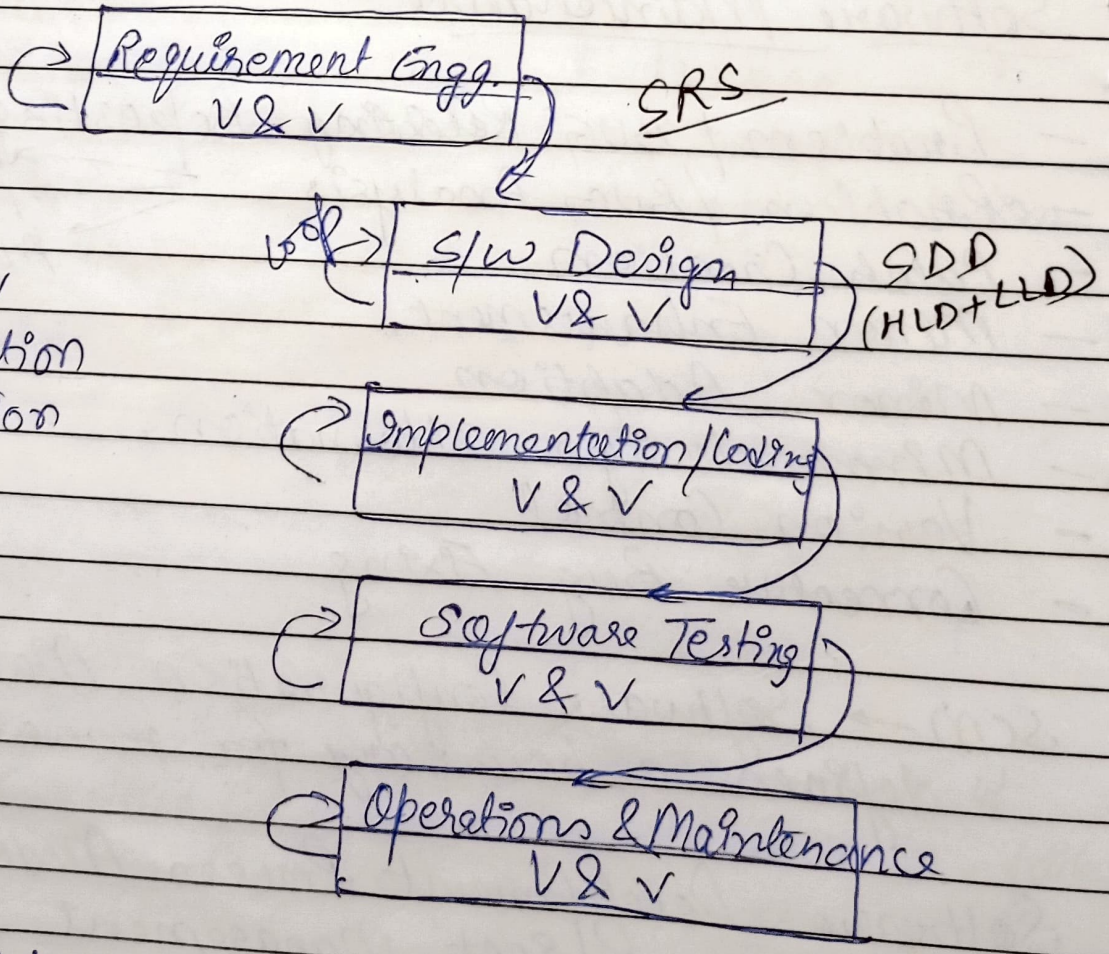
Date.....

1. LSM (Linear Shape Model) — Throwaway Prototype
2. PRM (Prototype Model) — Exploratory Protocol
3. V-Shaped
4. RAD (Rapid Application Development)
5. INM (Incremental Model)
6. Spiral Model (BSM) → Barrybohem
7. WIN-WIN Spiral Model
8. Agile Software development
   (Extreme, Pair, Scrub model, Agile.

* Ripple Effect should be 0.

LSM (Linear Shape Model). Sequential
Developed by 'Rote E' in 1990.

WATERFALL Model. (Stable & Static Requirements)



where V & V
is verification
& validation

* (0% Risk) because of Stable requirement

Date 24/3/22...

Waterfall model is the best software development for the beginners to develop small softwares where all the requirements of the software system are fully known.

Limitation of WSM

1) This model cannot be used in a projects where all the requirements are not known at the beginning of development.
2) In the waterfall model, we get the complete solution of the software at the end of testing phase. Then, only we can define satisfaction and dissatisfaction. If solution is dissatisfied (not good/ erroneous/incomplete/error-prone then it is shear wastage of time & cost & will lead failure.

Q. Describe Waterfall Model is Document-Driven Model.

* Waterfall Model is very systematic software development model in which every phase generates a specific document on the basis of that document, next phase get activated, following important document are generated by waterfall model by different phases.
1) SRS (Software Specification Document)
Requirement engineering phase generates this document which describes complete software requirement. On the basis of that

Spiral

Date.....

design phase is activated

ii) SDD (Software Design Document): ~~SDD~~
& Design phase produce SDD that explains or elaborate complete technical design of the software solution which is further decompose into component or module design named as high level design description document which also further decompose into several unit programs as low level design description document.

iii) Unit Test Plan & Unit Test Case Document

iv) User test Case & Test Suit Document.

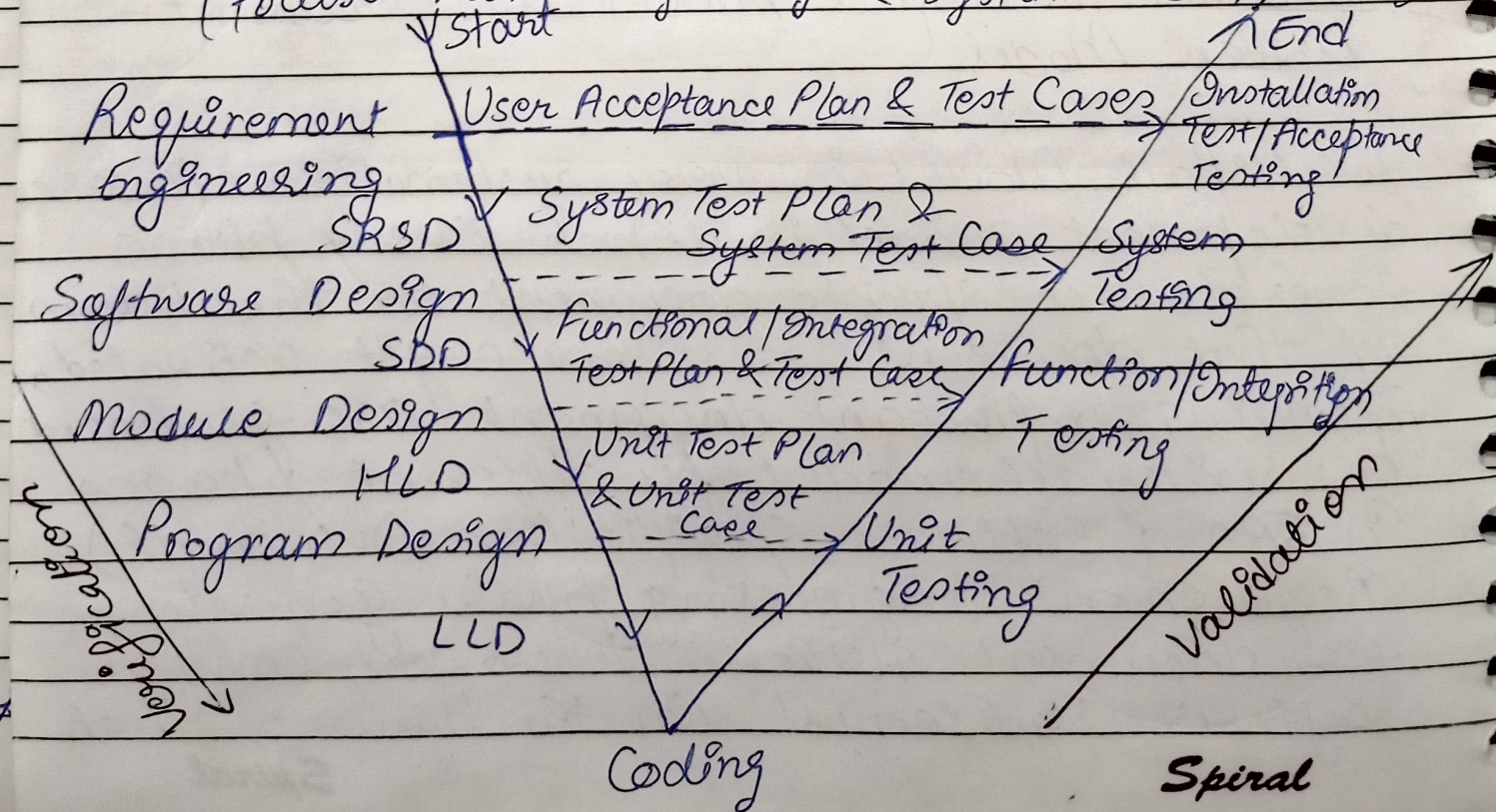
v) System Test Plan & System Case Document

vi) System Acceptance Test Plan & Acceptance Test Cases

vii) Installation Test Cases

viii) Maintenance & Enhancement Tracing Document

V-Shaped Model → Extension of LSM
(focuses more on Quality & Systematic Approach)

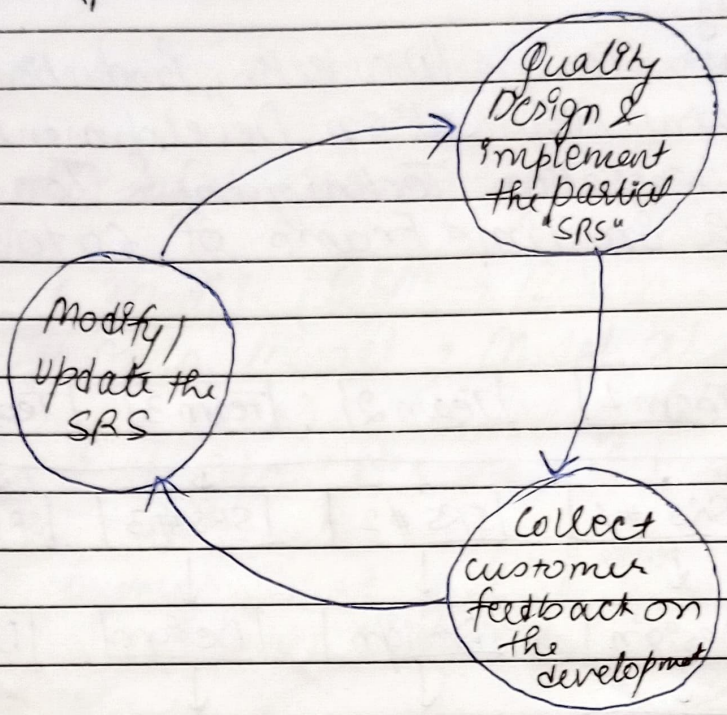


Spiral

Prototype → any working model

Date: 29/3/23

* Prototyping Model (PRM)



RAD (Rapid Application Development)

PROBLEMS

- Formal (IPO) — Input, Process, Output: known
  - Synthesis (problem with only one solution)
  - Analysis (problem with more than one solution & further decided)
- Informal — Input & Expected Output: known; Process: Unknown

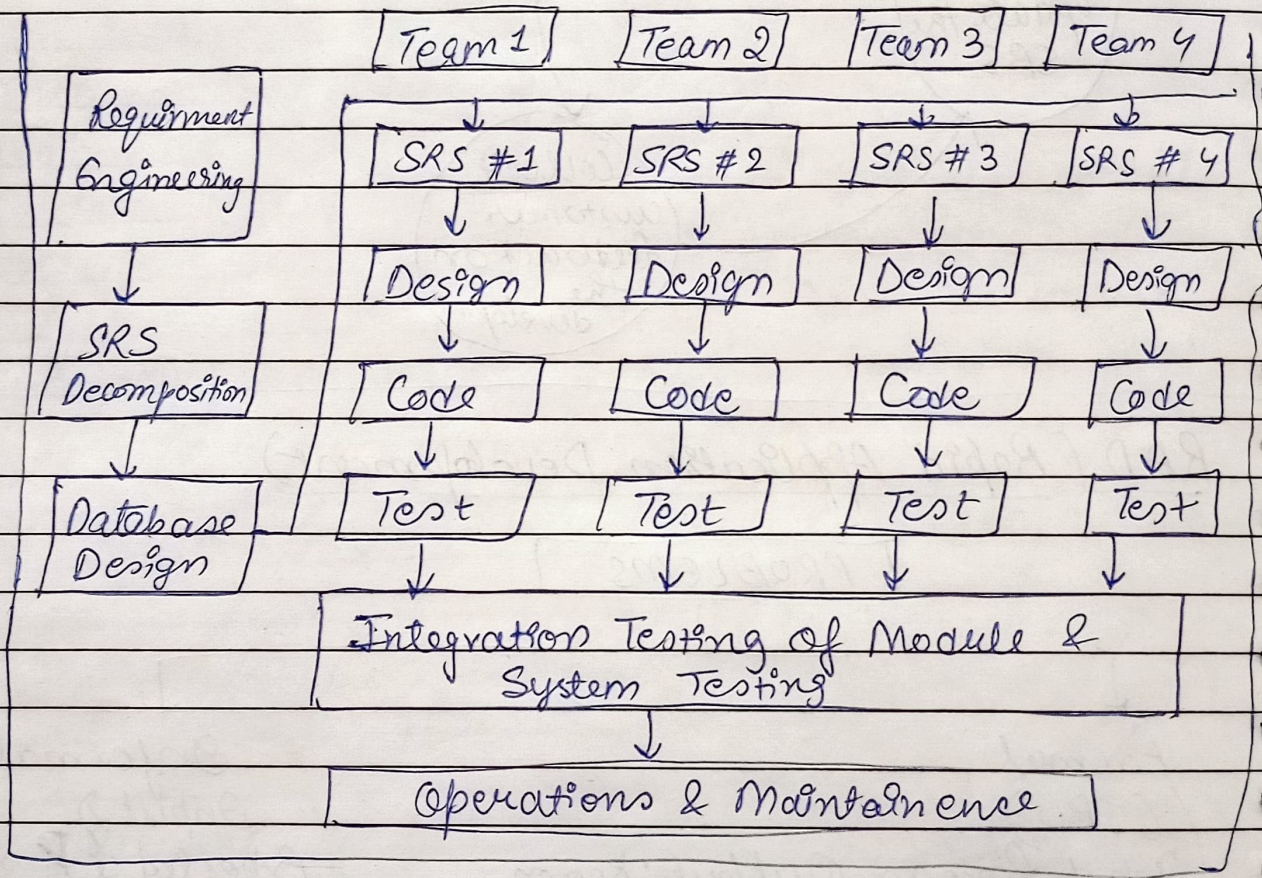
* Best Suitable for BUSINESS System & Expert Systems (Structured Solutions)

Spiral

Date 31/3/23

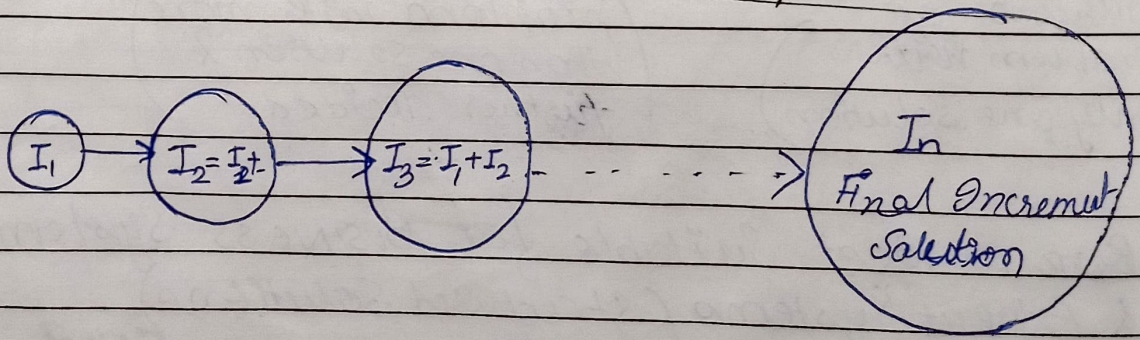
Component Based Model.

→ Reusability
→ Expert Sub-Teams (Marketing, Production, Sales)
→ JAD (Joint Application Development)
→ 4 GT (Generation Techniques & Tools)
→ Work With in Time Frame of 60 to 90 days.



Time frame of 60 to 90 days.

* Incremental Model / Phased Development Model



Spiral

Date.....

I ⇒ [Requirement Engineering] → [Design] → [Code] → [Testing] → [Incremental Module]

← Another Increment

* ~~In this we decompose~~

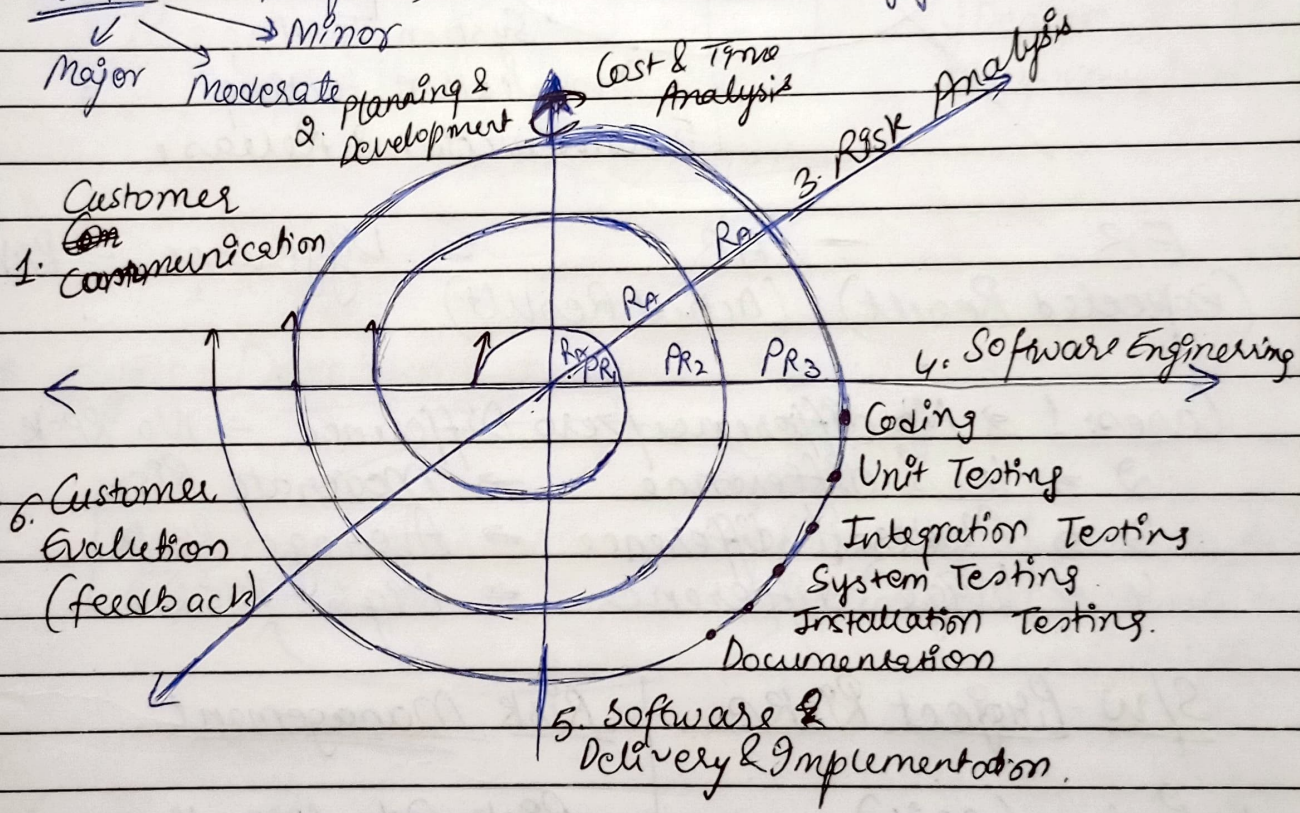
Spiral Model (BSM → Boheм's Spiral Model)

(Meta Model → Model about Models)

(69%) (18%) (13%)

Risks → People, Process, Technology

Major → Moderate → Minor



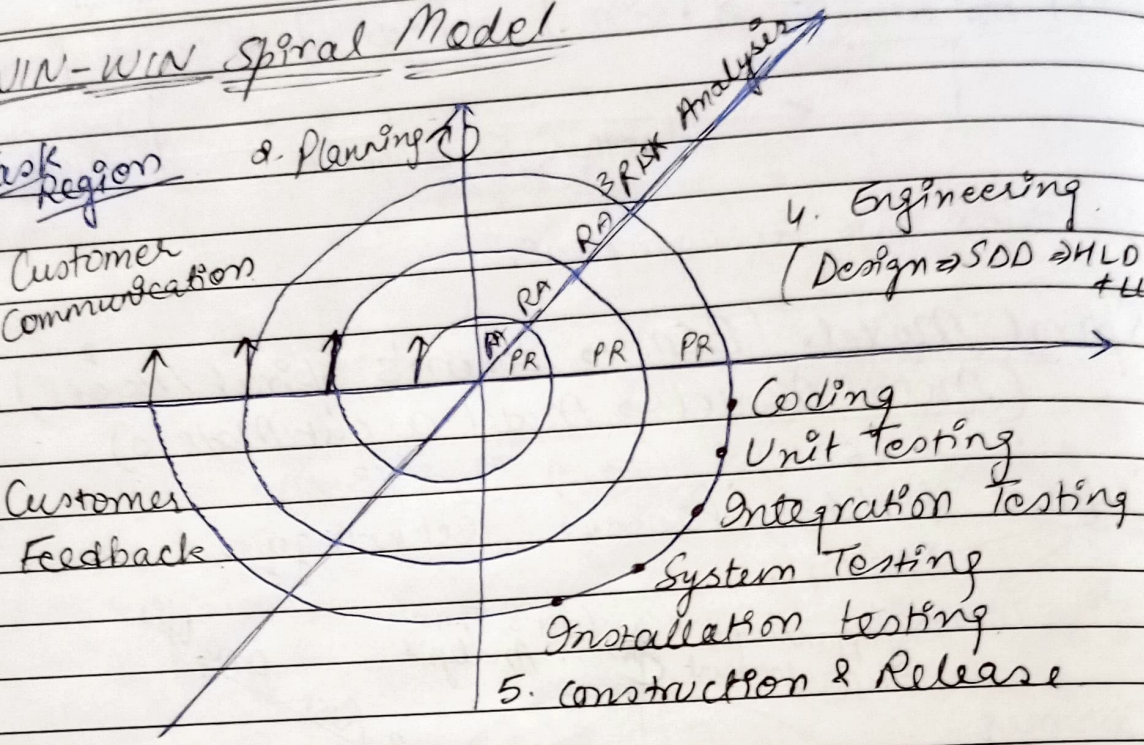
Note :- This model is best suitable for large & High Risk Software

* Peer Programming, Agile Computing, Extreme Programming, Scurb Model.

Spiral

Date... 5/4/22 ....

WIN-WIN Spiral Model



ER (Expected Result) − AR (Actual Result) = Difference → Risk

Cases: 1 ⇒ No difference / zero Difference → No Risk
2 ⇒ less difference → Moderate Risk
3 ⇒ Moderate difference ⇒ Average Risk
4 ⇒ large difference ⇒ High Risk

S/W Project Risks

1. People (69%)
2. Process (18%)
3. Technology (13%)

Risk Management

1. Risk Identification
2. Risk Assessment
3. Risk Planning
4. Risk Plan Execution + Monitoring & Control
5. Risk Mitigation

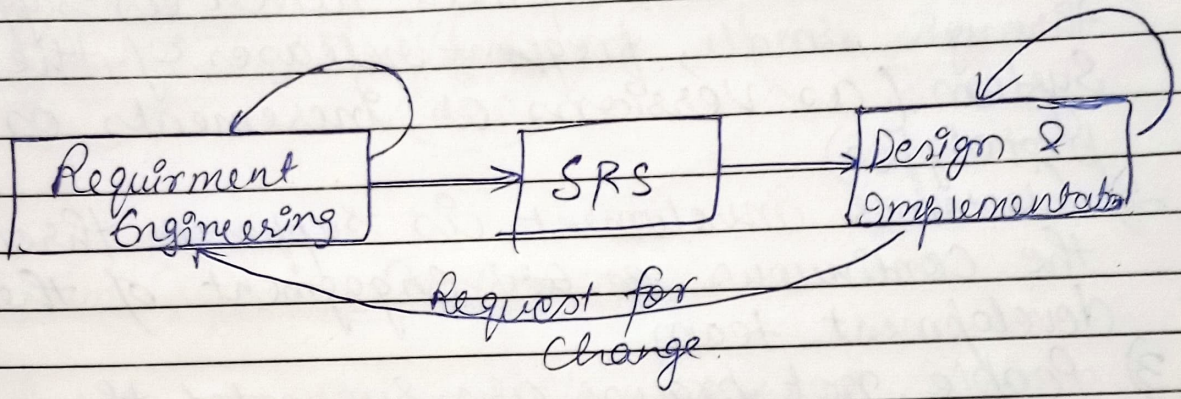
Spiral

Date.... 6/4/23

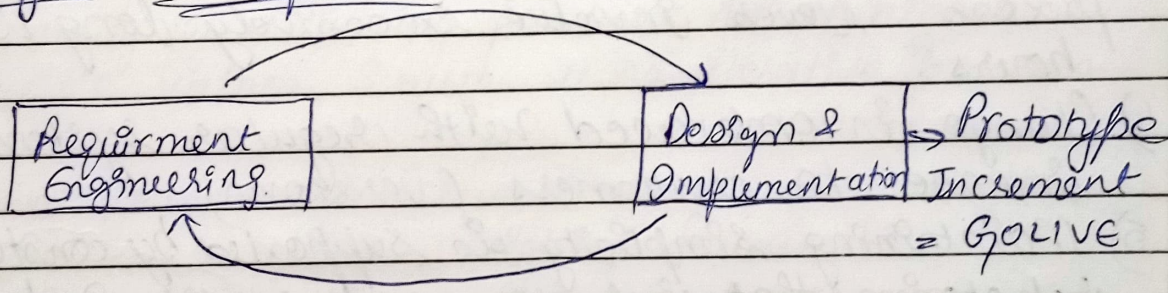
Strategies for Risk Resolve

1. Risk Collection
2. Risk Prevention
3. Risk Transfer Strategy
4. Risk Avoidance

Plan Driven & Agile Development



* Agile Development



Extreme Programming

It is the best known & ~~best~~ most widely used agile method. It was defined in 2000 by Beck. The approach of software development by pushing recognised, good practices such as iterative development to extreme levels. Extreme Programming, requirements are

In Spiral

Date.....

expressed as scenarios, which are implemented directly as a series of tasks. Programmers work in pairs & develop the tests for each task before writing a code.
Extreme programming involves a no. of good practices known as principles of Agile methods -

1) Incremental Development which is supported through small, frequent releases of the system (as versions or increments or prototypes)
2) Customer involvment is supported through the continuous ~~xx~~ engagment of the development team.
3) People, not process are supported through pair-programming, collective ownership of the system, code & a sustainable development process. Never involve excessively long working hours
4) Change is embraced with regular system release to customers (versions)
5) Maintaining simplicity is supported by constant refactoring that improves code quality & by simple designs

~~Role~~ scenario
* Select user stories for this release.

expressed as scenarios, which are implemented directly as a series of tasks. Programmers work in pairs & develop the tests for each task before writing a code.

Extreme programming involves a no. of good practices known as principles of Agile methods -

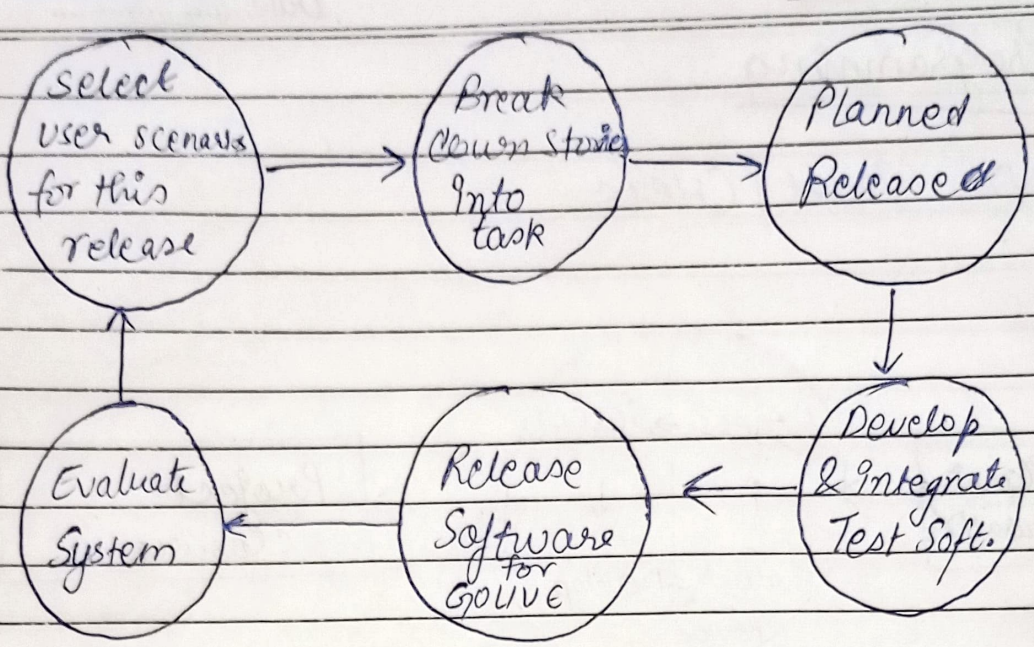
1) Incremental Development which is supported through small, frequent releases of the System (as versions or increments or prototypes)
2) Customer involvement is supported through the continuous ~~xx~~ ~~art~~ engagment of the development team.
3) People, not process are supported through pair-programming, collective ownership of the system, code & a sustainable development process. Never involve excessively long working hours
4) Change is embraced with regular system release to customers (versions)
5) Maintaining simplicity is supported by constant refactoring that improves code quality & by simple designs

~~Xt Rete~~ scenario

* Select user stories for this release.

XP Release Cycle.

Date.....

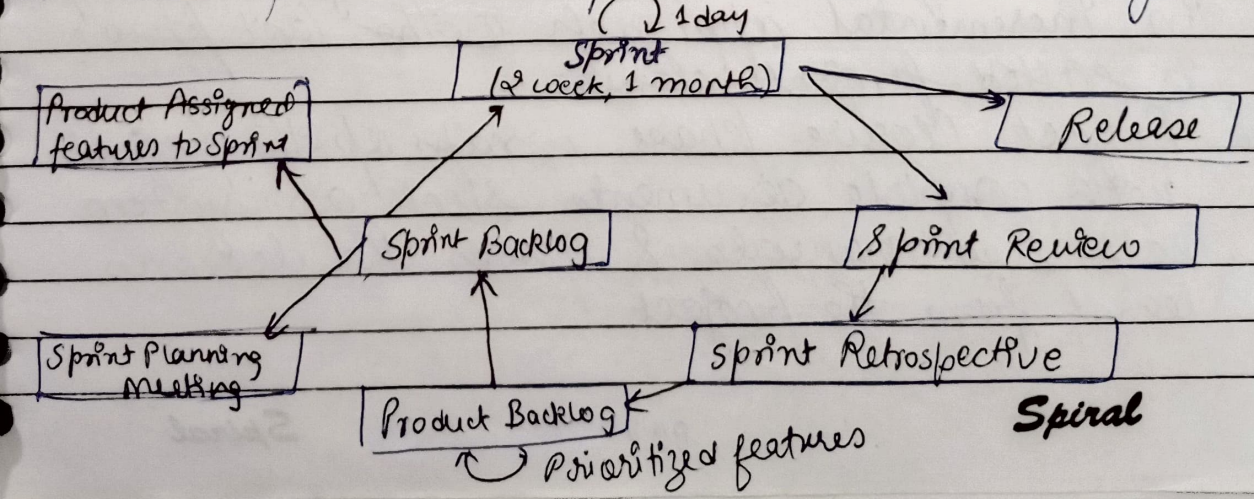


Scrum (Software Development)

Scrum is the type of Agile Framework. It is a framework within which people can address complex adaptive problem while productivity & creativity of delivering product is at highest possible values. Scrums uses Iterative process.

Silent features of Scrum are:

1. Scrum is light-weighted framework.
2. Scrum emphasises self-organisation
3. Scrum is simple to understand
4. Scrum framework help the team to work together.



Spiral