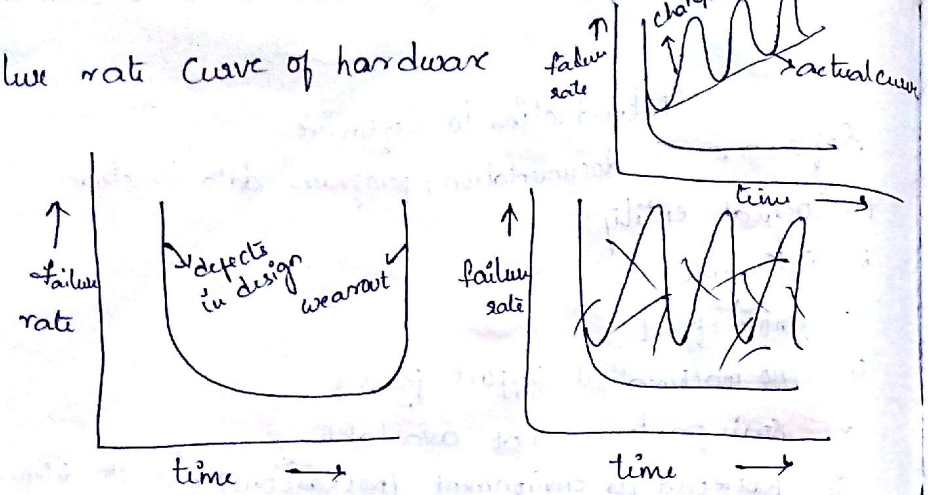
Software: Software is defined as

1. Set of instructions (computer programs) that when executed provide desire features and functions
2. data structures that enable the programs to manipulate information
3. Documentation that describe the operation and usecup the programs

Characteristics of software:

1. Software is logical rather than the physical system element, it is itangible
2. Software is developed or Engineered, it is not manufactured.
3. Software does not ~~wear~~ wear out

failure rate curve of hardware



Although the industry is moving towards Components based Construction, more software Continous to be Customized

Types of Software: (or) Changing nature of Software:

1. System software
2. Application
3. Engineering & Scientific
4. Artificial Intelligence
5. Embedded
6. product
7. webapps
8. (distributed) Ubiquitous software
9. Netsourcing
10. Open Source.

Role of the Software:

Software takes on a dual role. it is both product and vehicle for delivering the product

- As a product it delivers the computational capabilities by the network of computers. whether software resides within a phone or operates inside a Computer it is an information transformer - Producing, managing, acquiring, modifying, displaying or transmitting information that can be simple as single bit or as complex as multimedia presentation.

→ As a vehicle for delivering the product Software acts as basis for "Control of Computer.

Ex: OS.

Legacy Software:

Legacy Software is older software, developed decades ago and still working properly.

The quality of the legacy software. The main Characteristics of the legacy software is poor quality because

1. legacy Systems have inextensible designs
2. Convoluted Code (Complex code/complicated)
3. poor (or) Non-Existent documentation.

and at These Systems support "four" business functionalities

The legacy systems evolve for the following Systems

1. The Software must be evaluated to meet the adaptations needs of the new business environment
2. The Software must be enhased to adopt the needs of the new Computing environment
3. The Software must be extended to make it interopable with more modernt systems (or) data bases
4. the Software must be rearchitected to make it lively Succesfull within a network environment

Software Myths:

Software Engineering
approachable develop
software
approach to develop
tools & process

1. Management Myths
2. Customer Myths
3. pactioners Myths

1. Management Myth

1. All the contents and the description are available in books.
   No, if is wrong

2. If the time is not sufficient to complete project. they take more employees to appoint to the project work

3.

Customer Myths

1. Software is flexible.

2.

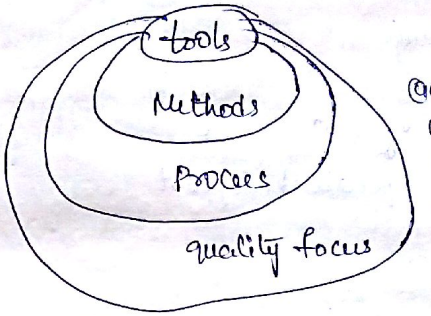
practitioner Myths

1. One statement of idea / code is enough to complete project
2. No change to modify untill the product is out

3.

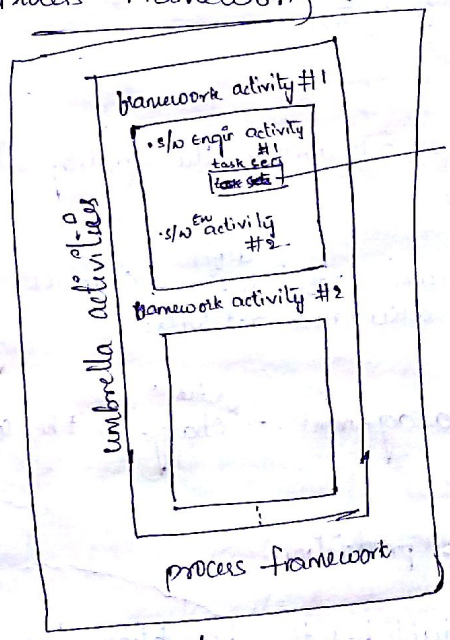
Generic view of Process

sequence
disciplana
Quantifie



deployment
@Construction
③ Planning
② Modeling
① (Discussing) communication

Process Framework (road map)



work tasks
work product
work quality assurance points
milestones

S/W process

each process encompasses sets of :

Communication → requirements gathering

Planning → ① risk management, ② Technical tasks to be applied ③ work products to be produced ④ schedule ⑤ Resources

(Modeling): Analysis - Better understanding of problem is analysis
Design — algorithm form of solution to problem
set actions

Construction: Coding
Testing — to finding the Errors

Deployment: product delivery to customer, feedback, support

Umbrella activities:

1. Project schedule tracking & control
2. Risk Management
3. Formal Technical Reviews - To ensure the quality
   To uncover errors
4. Software Quality Assurance - to define & conduct the assurance activities
5. Measurements
6. Software Configuration Management - related to change in the software
7. Reusability management
8. Work product preparation & production

1. A process framework establishes the foundation for a Complete Software process.
2. A process framework encompases small number of framework activities that are applicable to all software projects
3. In addition, the s/w process framework encompases a set of umbrella activities that are applicable across the entire software process.
4. Each framework activity is populated by a set of s/w engineering actions - a collection of related tasks that produce a major s/w engineering work products. each software engineering activities is populated with individual work tasks that accomplish some part of the work.

The Process framework encompases the following framework activities

1. Communication: This framework activity involves heavy Communication & collaboration with Customer & encompases requirements gathering
2. planning:

Process pattern: pattern - a template - software activities

The Software process can be defined as a collection of pattern the pattern defines a set of activities actions work tasks work products required to develop software.

The process pattern provides a template - a method for describing an important characteristics of the software.

1. pattern-name: A pattern is given a meaningful name that describes its function. from the pattern name one can guess its functions. Ex: Customer Communication
2. Intent: The objective of the pattern is described the intent of the customer communication is to establish a collaborative relationship with the customer

3. Type: The pattern type is specified here. There are 3 types of patterns 1. Task 2. stage 3. phase.

1. Task: It represents sc action or work task that is part of the process
   ex: requirements gathering

2. stage: It represents framework activity
   ex: planning, Communication

3 phase: It represents sequence of frameworks activities
   ex: Spiral model.

4. Initial Context: In this section, the conditions under which the pattern applied are describe. prior to the initiation of the pattern is described
   ex: planning pattern requires that 1. customer and sc have established a collaberative communication
   2. Complication of the number of task patterns for the customer communication of the pattern
   3. Basic requirements projects scope, project constraints known.

5. Problem: A problem to be solved by the patter is described
   Ex: Insufficient requirements is a problem.

6. Solution: Implimentation of the pattern is specified
   Ex: the above side problem has a solution. the solution is
   1. Establish effective Communication with the customer.
   2. Ask questions in order to obtain meaningful requirements

7. Resulting Context: It describes the results after successful implimentation of the pattern sufficient requirements are the results.

8. Related patterns: A list of process patterns that are directly related to the one are provided as a hierarchy
   ex: Communication stage pattern encomposes the task assembly requirement gallery, constraints description.

9. known uses: (or) Example: the specific instances (or) applications in which the described pattern is useful mentioned in the Section

CMMI (Capability Maturity Model Intigration)

The Software Engineering Institute (SEI) is developed a process model that emphasises the process maturity. It is predicated on a set of system and software engineering capabilities that should be present when the organisations reaches different levels of process capability and maturity

various capability levels are.

level 0 level 1 — level 2 level 3 — level 4 level 5

0 Incompleted 1 performed — 2. Managed 3. defined — 4. Quantitatively Managed 5. optimised

CMMI represents a process meta model in two different ways as a 1. continous & 2. staged

In continous model CMMI defines each process area in terms of specific goals and practices required to achieve them goals.

The staged CMMI model defines the same process areas, goals and practices as the continous model.

The primary difference is that the staged model defines five maturity levels rather than capability levels.

// P.ASSESSMENT

Personal & Team Software Process:

Software
Software Engineering
Myth
legacy Software
Communication

Process assesment
Process
Construction
modeling
deployment

Personal & Team Software Process

PSP needs the needs of the individual, team, Orgnization

1. planning
2. High level design — component
3. " " " Review
4. Development
5. post Mortem

TSP
self directed team

1. (Communication) launch
2. (planning)
3. High level design
4. Implementation & test
5. Integration & test
6. Postmortem

PSP:
Each S.E would create a software process that best fits his needs and at the same time meets the needs of the team and the organization

TSP:
A Team itself Creates its own process it meets the needs of the team and at the same time meets the needs of the individuals and needs of Organization

Psp:
Psp defines five framework activities
Process

1. planning: This activity isolates requirements & develops both & estimations &
   2. All metrics are recorded
   3. finally development tasks are identified and schedule is created.
2. High level design: Component design is Created. prototype are build when uncertainity exists.
   2. All issues are recorded & tracked.
3. High level design review:
   Formal verification methods are applied to uncover errors in the design.
   2. Metrics are maintained
4. Development:
   The Component level design is refined and reviewed, Code is generated. reviewed, compiled and tested.
   Metrics are maintained for all important tasks
5. Post Mortem:
   using methods & metrics collected, the effectiveness of the process is determined. These provides guidnes for modifying the process to improve its effectiveness
   PSP represents a discipline, metric based approach to software engineering.

TSP:
The goal of the TSP is to build the self directed Project team that Organize itself to produce high quality Software.

Objectives of TSP:

1. Build self directed team (3 to 20 members for teams) managers should coach and motivate the team members.

2. framework activities:

TSP defines 1. Launch
2. High level design
4. Integration & testing
3. Implimentation
5. PostMortem

Launch: Tsp make the use of scripts, forms and standards provide guidens to team members in their work.

PostMortem: Sets the states for improvement

implementation: Code generation & Testing

Integration: Integrate all the components & test the Software to uncover errors

High level design: Component level design.

Communication

Construction: The frame work activity involves heavy communication with the customer and encompases requirements gathering and other related gathering activities.

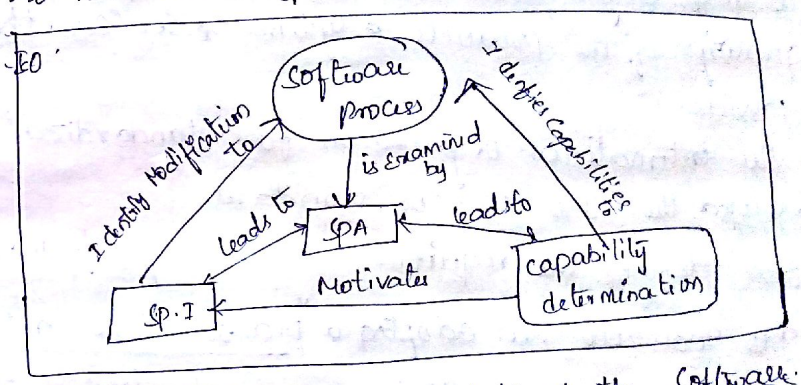
legacy Software:

***TSP

Software process assesment:

attempts to understand the current state of the process and to (implement) improve the process model.

1. SCAMPI (standard
2. CBA IPI
3. SPICE
4. ISO 9001:2000 (plan-do-check-Act)
5. SEO



It attempts to understand current state of the Software Process with the intent of improving it.

Approaches for Assesing the S/W process:

1. SCAMPI (standard cmm Accessment method for process Improvement):

provides five step process assessment to assess the process

-This uses SEI cmm as a basis for the assesment

cmm based aprisal for inlinal process improvement: (BAIPI

Recall:

1. It provides a diagonolistic technique for process assesment.

2. This uses SEI CMM as a basis for assesment.

3. SPICE : Spice standard defines set of requirements for Software process assesment.
The intent of the standard is to assest the organization in developing an objective of the Software process.

4. ISO-9001:2000 : It is a generic standard that applies to any Organization that wants to improve the overall quality of the products & Systems & Services that it Provides.

→ The international organization for standardization has developed the ISO-9001:200 standard.

→ It defines the requirements for the quality management S/w

→ ISO-9001:2000 has adopted a 'plan-do-check act' cycle that is applies to quality management elements of the S/w project.

→ In this plan - establishes the process objectives activities and tasks that are necessary to achieve high quality Software and resultant Customer Satisfaction

→ do - implements the process
- Check - monitors and measures the process &
- act - initiates the process improvement

(Turn page to back diagram)

SPA - Software process Assesment
SPI - Software process Improvement

Process technology:

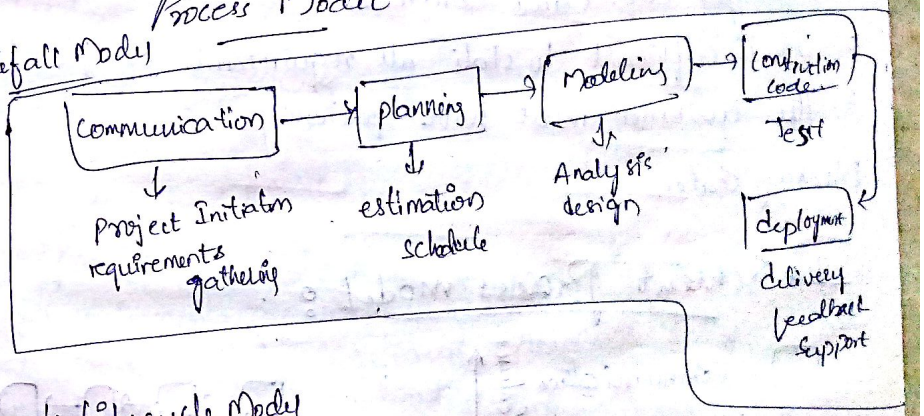
Process technology tools have been developed to help S/w Organizations.

1. Analyse the Current process
2. Organize work tasks
3. Control and moniter the progress and & manage technical quality

Product and process :

If the process is week then the end product will suffer

Process Model



1. Lifecycle Model
2. Linear
3. Systematic approach

Prescriptive Process Models :

1. PPM dose prescribe the set of framework activites, Software engineering actions, tasks, workproducts, quality assurance activities and change Control Management that are required to develop high quality Software

2. It also prescribes a work-flow - i.e., the manner

in which the process elements

Waterfall Model:

It also called as classic lifecycle model. It is Systematic Sequencial approach.

⇒ Requirements of the problem are well understood from

⇒ Well defined adoptation or enhancement of the existing System

turn back Diagram

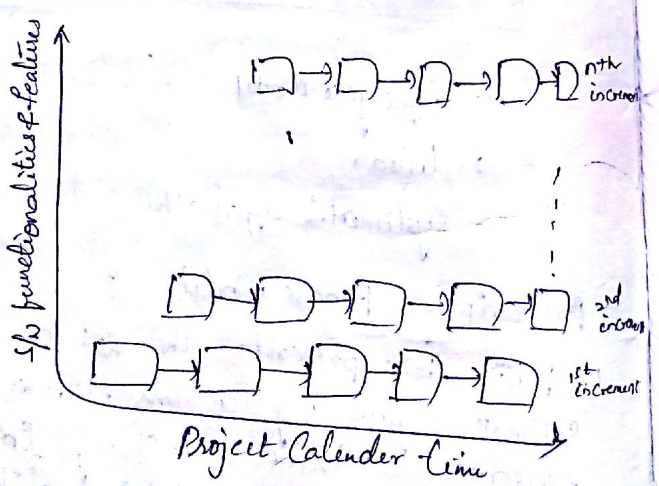
Drawbacks:

1. Changes can Cause Confusion
2. It is difficult to state all requirements at first time
3. The customer must have patiency

Blocking states

Increment Process model:

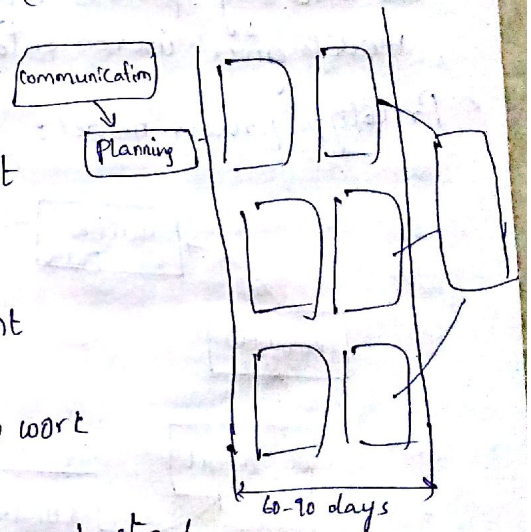
- communication
- planning
- modeling
- Construction
- deployment



in an iterative fashion

- Each linear Sequence produce an increment
- First increment is the core product
- The Core product is modified to better meets the needs of the customer
- It is useful when Sufficient staff is unavailable

Rapid Application Development (RAD):



- It uses the elements of the water fall model
- uses the short development cycle
- Rapid development was achieved by using component based construction
- It uses the five frame work components
- If requirements are well understood and process scope is constraint RAD create fully functional System within short period (60-90) days)

Drawbacks:

1. Large Scale projects requires sufficient human resources for RAD teams
2. If developers and customers are not communicate for Rapid for activities, RAD models will failed.
3. Proper modularization is must.
4. Not Suitable, if technical risks are high.
5. If high performance is an issue then RAD is not Suitable

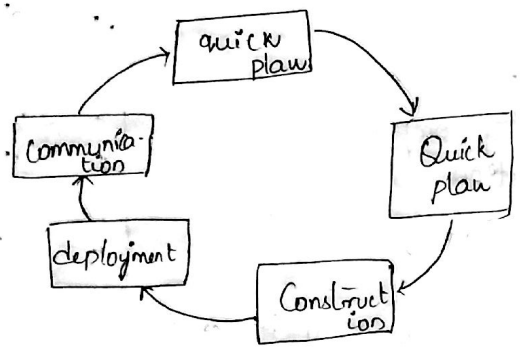
SM

Evolutionary Process models :

- Software evolves over a period of time because business and product requirements change as development proceeds. In this situations software engineer need a process model that has been explicitly designed to accomdate a product that evolve over time
- evolutionary process models are iterative. These models gives incremental versions of the s/w.

Prototype process model :

requirements fluffy



⇒ In Communication, phase Customers and talk to each other to fullfill the requirements of the customer and known the objectives of the customer

⇒ proto type process model is used in some other process model

⇒ It is And the areas where these are used

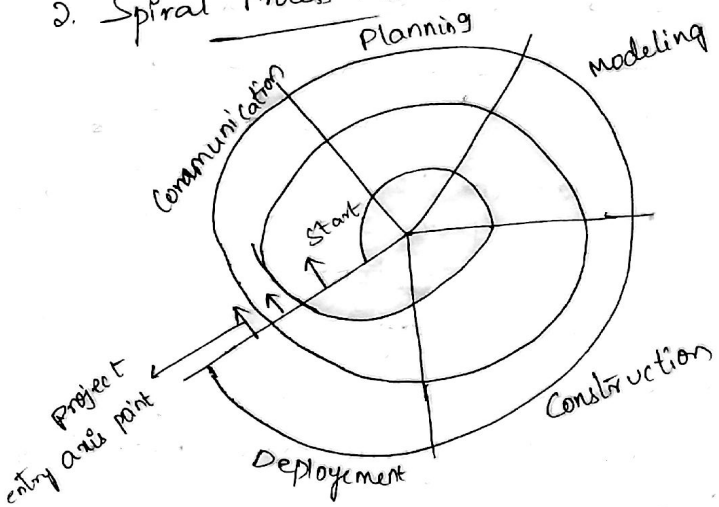
⇒ After communication, Quick plan that leads to Quick design ie., input, output format.

⇒ After design they develop the prototype

⇒ prototype is not an operational Software. And prototype is used to fullfill the requirements

⇒ After Analysis, the feedback of the customer used to develop

2. Spiral Process Model :



⇒ It couples the iterative nature of the prototype in and controlled and systematic Aspects of the waterfall model.

⇒ It provides Rapid development

⇒ using Spiral model, Software is developed in series of evolutionary releases.

⇒ During early iterations, the release might be a prototype.

⇒ During the later iterations increasingly more

Complete Versions of the Software are produced.

⇒ The Spiral model is divided into set of frame work activities defined by the team

⇒ Each of the framework activities represents one segment of Spiral path.

⇒ It is best approach for the large scale systems because Customer & developer can better understand the problem at each evolutionary level also, risks identified and rectified at each level.

⇒ In the initial ~~paths~~ pairs prototype is developed and then more improved versions of s/w Software is developed.

⇒ during planning phase the cost and schedule of the software can be plan and adjusted based on the feedback from the customer.

⇒ In Spiral model project entry point access is define

⇒ This access dog represents starting point for each evolutionary paths. The spiral model uses prototyping as a risk reduction mechanism

Advantages:

1. Requirement change can be made at every stage
2. risks can be identified and rectified before they get problematic

Drawbacks:

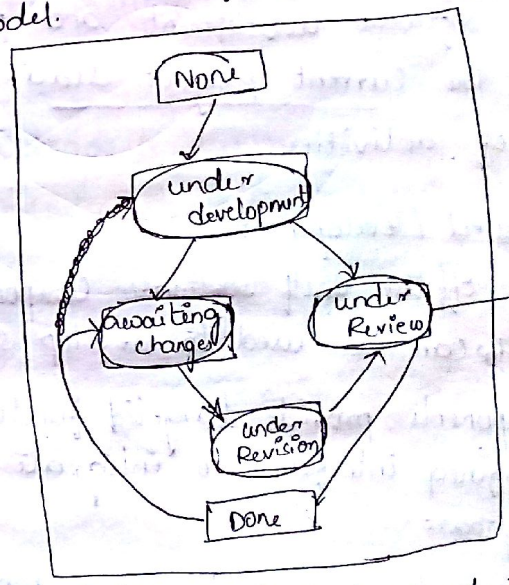
1. Demand considerable risks assesment.

Concurrent Process Model:

⇒ The Concurrent development Model Sometimes called Concurrent Engineering can be represented as series of framework activities, software engineering actions, tasks and their associated states.

For Ex: modeling activity is accomplished by analysis and design.

⇒ The following figure provides the systematic representation of one software engineering action with in the modeling activity for the Concurrent Process model.



⇒ the activity may be in any one of the state at any given time. Similarly, other activities or tasks can be represented in the same manner.

⇒ All the activities exist concurrently but resides in different states.

For Example: In a project, the communication activity has completed and its first iteration and awaiting changes stati.

⇒ The awaiting activity which existed in the none stati while planning completed now makes a transition in to under development.

⇒ If however, Customer makes any changes in the requirements, the modeling activity changes to awaiting changes stati.

⇒ The Concurrent process model is applicable to all types of Software development and provides accurate picture of the current project stati. It defines network of activities.

Component Based Development:

⇒ Commercial off the shelf software components developed by the vendors can be used to develop software.

⇒ These components provide targeted functionalities with well defined interfaces to integrate the components in to software.

⇒ The Component based developed model incorporates many of the characteristics of the spiral model. It is evolutionary nature of uses iterative approach to create the s/w

⇒ The model composes applications from the s/w Components

⇒ Modeling and Construction activity begins with the identification of the codidate Components

⇒ The Components based com it incorporates the following steps

1. available Component based products are researched and evaluated
2. Component integration issues are considered
3. A s/w architechture is designed to accomdate the Components
4. the components are integrated in to the architecture
5. Testing is Conducted to ensure proper functionality

Merits:

1. This process leads to s/w reuse
2. This process leads to 70% reduction development time
3. 84% reduction project cost.

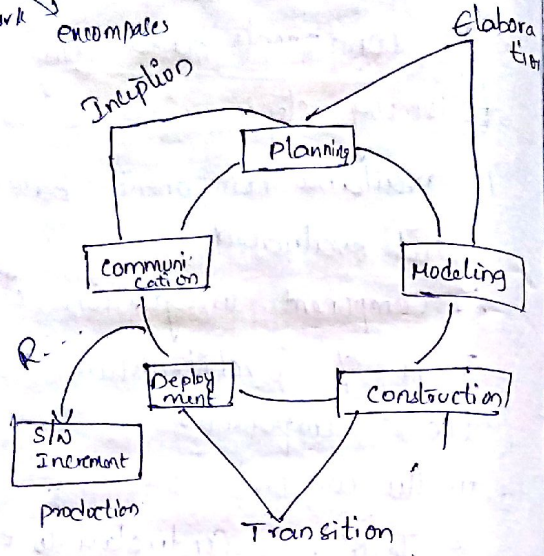
Unified Process Model: UP (or) RUP used to develop object-oriented

UML (users) provides framework encompasses s/w system

Interactive increment

5 framework activities

1. Inception
2. Elaboration
3. Construction
4. Transition
5. Production



Inception – 1. Requirements Identification
2. Plan → resources, risks, schedule
& rough architechture – outline of the sub system
4. preliminary Use-case.

Elaboration – 1. use-case – Refined
2. Expand → use-case
Analysis model
Design
Implimentation
Deployment model.
& plan is renewed
↓ modify if necessary

– Unified Process model provides framework to develop the (conventional generic software

– unified process Model provides the framework to develop object oriented s/w system where as the generic process

1. Inception : It encompases both customer communication and planning activities
- By colabarily with customer and endusers requirements for the s/w are identified.
- Ruf architecture of the system is proposed
- plan ofs the project is developed.
- Requirements are described through a set of use-cases
- In general use-case describe sequence of systactions that are performed by an actor as the actor interact with the s/w.

- Use-cases helps to identify the scope of the project and provides a basis for project planning.
- Architecture at this point is nothing more than the outline of the major subsystems. Later the architecture will be refined.
- Planning identifies resources ss major risks and defines a schedule.

2. Elaboration : The Elaboration activity encompases communication and modeling activity of the generic process model
- Elaboration refines the priliminary use-cases is known as refinement.
- It expands the architectural representation to include 5 different views of the system
1. use-case model 2. Analysis model 3. Design M
4. Implimentation M 5. Deploy Model.

- In some cases, elaboration creates a "executable operational base line" that represents executable system
- In addition plan is reviewed to ensure that scope risks and deliverable data remain resonable.
- Modifications to the plan may be made at this time

Construction: const phase of the UP is identical to the Construction activity defined for generic software
- using architectural and the model input the Construction phase develop software components that we may use each case.

Transistion:

In this phase up encompasses the later stages of Construction activity and 1st part of generic deployment activity

s/w is given to end users and user feedback reports both defects & necessary changes.

In addition s/w team creates a necessary support that is required for release.

At the conclusion of the transistion phase the s/w increment become a usable s/w release.

Production:

The production phase of up coincides the development activity of the generic process

2. During the phase the ongoing use of s/w is monitered, support for the operating environment is provided.

Agile process model:

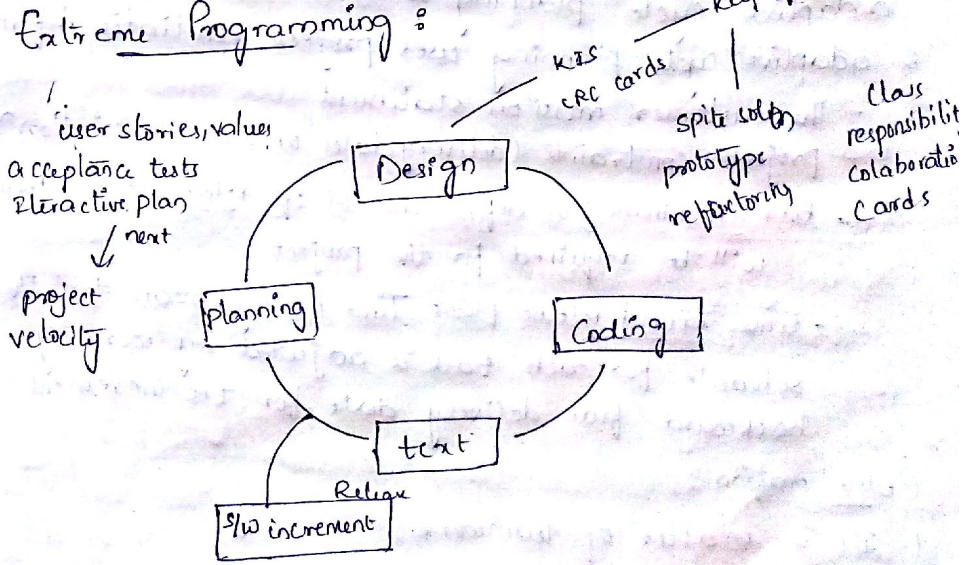
What is Agility?

1. effective response to change.
2. Effective response to change. Communication among all stake holders
3. Drawing customer on the team
4. self organising teams
5. yeilds rapid increment delivery of s/w.

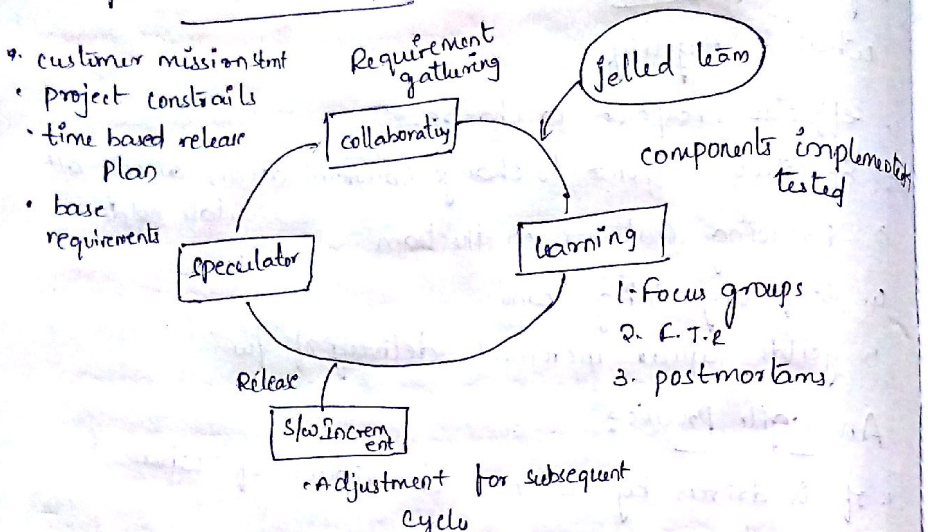
An agile process:

1. It is driven by customer description of what is required
2. Recognises the plans or short live.
3. Develops s/w iteratively with heavy enpasses on Construction activity.
4. Delivers multiple s/w increment
5. Adapt as changes occur

Extreme Programming:



Adaptive S/w Development (ASD)



- It is a technique for buiding complex s/w systems
- ASD life cycle incorporates
  1. Speculator 2. collaborating 3. Learning

1. Speculator :
   1. During speculation the project is initiated and adaptive cycle planning is conducted.
   2. adaptive cycle planning uses project initials information
      - The customer mission statement
      - Project constraints (delivery date or user description)
      - basic requirements to define the set of release cycles (s/w increment) that will be required for the project
      - time bound release plans Time boxing means that the schedule for each task is adjusted by working backward from delivery date for the increment

Collaboration :
1. It is matter of teamwork
2. jelled team is essential for real collaboration
   a jelled team is a group of people so strongly knit together that the whole is better than the some of, parts
3. It is a matter of trust
4. people work together must trust one another to
   - critisized w/o any mocity
   - Assert w/o angry (ressertment)
   - work hard as hard as possible
   - Has skill set to contribute to the work

Learning:
1. ASD encompass on learning
2. learning will help the team to improve their level of real understanding
3. ASD team learns in 3 ways
   - Focus groups
   - formal technical Reviews (FTR)
   - post Mortem
4. Compliments are implemented and tested in this phase.

Scrum

- It is an assaignment process mody
- Scrum principles are
  1. small working teams are organised to "maximize communication, minimize overhead and maximize sharing of knowledge.
  2. The process must be adaptable to both technical and business changes.
  3. The process it yields frequent software implements
  4. Development work and people who performed in are partition into low coupling partition or packets.

5. Constant testing and documentation is performed.

⇒ Scrum process model incorporates the following framework activities

1. requirements
2. analysis
3. Design
4. Evolution
5. Delivery

within each framework activity worktasks occur within a process patting called Sprint

⇒ Scrum Emphases the use of set of process patterns. Each of these process patterns defines a set of development activities.

Backlog: It is A prioritised list of project requirements or feature that provide business value for the customer.

2. Items can be added to the backlog at any time
3. The product Manager asserts the backlog and updates priorities as required

Sprints: Sprint Consists of work units that are required to achieve a requirement defined in the backlog.

2. Sprint allows pr team members to work in a shortterm but stable environment

Scrum Meetings: Scrum meetings are short meetings held daily by the scrum team

- three questions are asked and answered by the team members

1. what did you do since the last meeting?
2. what obstracles are encountaring?
3. what do you plan accomplish by the next meeting

A team leader called Scrum master leaves the meeting and assesses the responses from each person. The Scrum meeting helps that team to uncover problems as early as possible.

Demos: Demos delivered the software increment to the customer so that functionality that has been implemented can be demonstrated by the customer.