

# UNIT - 1

## LAOs

### ① a) Changing nature of Software

Seven broad categories of computer software present continuing challenges for software engineers:

#### 1. System Software:

A collection of programs written to service other programs & the software used by systems.

Some system software processes complex but determinate, information structures.

Ex: compilers, editors & file management utilities.

Other systems like application process largely indeterminate data.

Ex: Operating System Components, drivers, networking s/w.

#### 2. Application Software:

Application software is defined as programs that solve a specific business need.

In addition to convention data processing application, application s/w is used to control business function in real time. Ex: real-time manufacturing process control, ms-office, air traffic control.

#### 3. Engineering and Specific Software:

This s/w is used to facilitate the engineering functions and tasks.  
simplify

Ex: Computer-aided design, Cadcam, matlab, auto cad...



#### 4. Embedded Software:

Embedded S/w resides within the system or product and is used to implement and control feature and function for the end-user and for the system itself.

Ex: Digital functions in an automobile such as fuel control, dashboard displays, and ~~break~~ braking systems.

#### 5. Product-line S/w:

Designed to provide a specific capability for used by many different customers, product line software can focus on the limited and esoteric marketplace or address the mass consumer market.

Ex: Word processing, spreadsheets, computer graphics.

#### 6. Web Applications: It is a client-server computer program which the client runs on the web browser.

Ex: Google doc's, Microsoft live, web pages...

#### 7. Artificial Intelligence S/w:

It makes use of non-numerical ~~also makes~~ algorithm to solve a complex problem that is not amenable to computation or straight forward analysis.

Ex: Robotics, game playing, pattern recognition...

#### New challenges

1. Ubiquitous Computing: wireless networks
2. Net sourcing: web as a computing engine, cloud
3. Open world computing: Distributed computing
4. Open ~~world~~ source computing: Ex: "free source open to computing community".

#### b) Evolving role of Software

##### ...1. A Product

##### 2. Vehicle

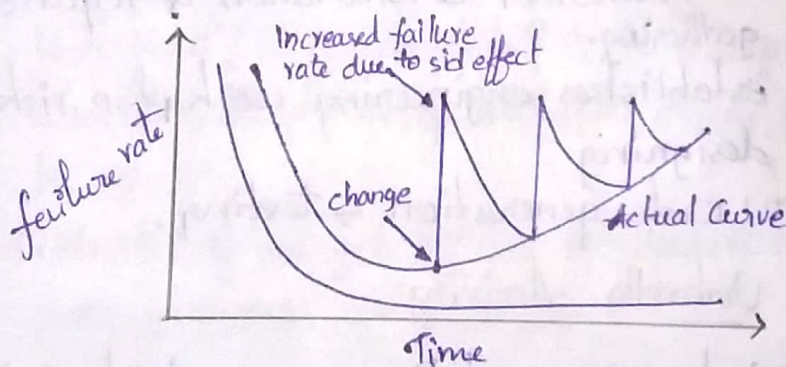
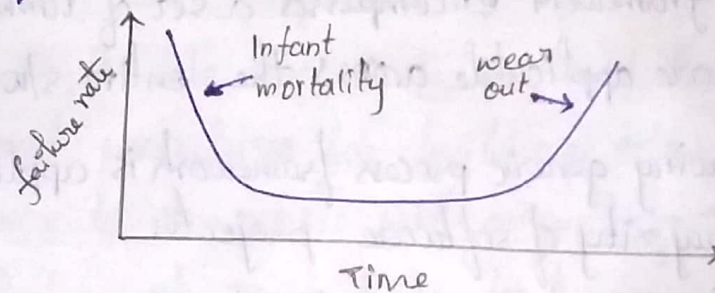
If it is for 5 marks write about Software, how it is developed and points 1. and 2.



- 2) Software: Software is a set of instructions, data or programs used to operate computers and execute specific tasks.

### Software characteristics

- > Software is developed or engineered, it is not manufactured in the classical sense.
- > Software doesn't "wear out".



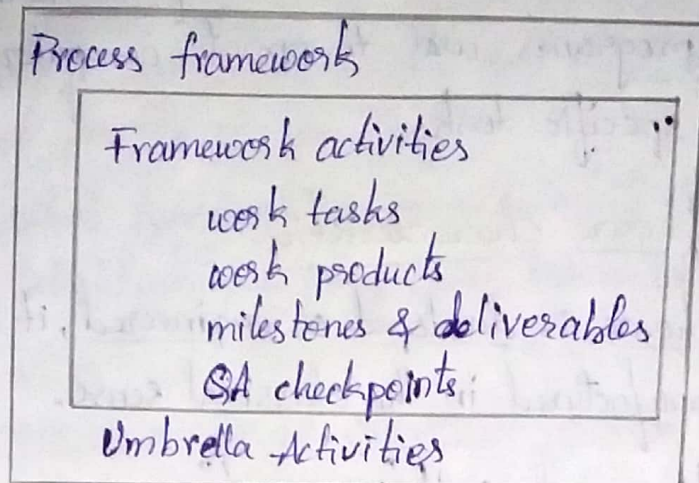
- > Most s/w is custom-built, rather than being assembled from existing components.

### 3) A Process Framework

A process framework establishes the foundation for a complete s/w process identifying a small number of framework activities that are applicable to all s/w projects, regardless of their size or complexity.



## A common Process Framework



The process framework encompasses a set of umbrella activities that are applicable across the entire s/w process.

The following generic process framework is applicable to the vast majority of software projects:

- > communication: customer collaboration & requirement gathering.
- > planning: establishes engineering work plan, risks
- > modelling: designing
- > construction: code generation & testing.

### Umbrella Activity

- > Software Project Management: Progress against project plan and actions.
- > Formal technical reviews: remove errors before going to next action.
- > Software quality assurance: Conducts the activities required to assure quality.
- > Software configuration management: Manages the efforts of change throughout the process.
- > Document preparation & production: Prepare required documents.
- > Reusability management: Product reuse.
- > Risk management: assess the risk that affects the outcome.



#### 4) Software Myths

Software myths propagate false beliefs and confusion in the minds of management, users and developers.

Many software problems arise due to myths that are formed during the initial stages of software development.

There are 3 types of software myths.

##### 1. Management Myth.

1. Myth: We already have a book that's full of standards and procedures for building software. Won't that provide my people with everything they need to know?

Reality: The book of standards may very well exist, but is it used? It is complete? Is it adaptable to your situation, deadlines, quality? In many cases, the answer is no.

2. Myth: If we get behind schedule, we can add more programmers and catch up.

Reality: > Software development is not a mechanistic process like manufacturing.

> Adding people to a late slow project makes it later.

3. Myth: If I decided to outsource the slow project to a third party.

Reality: If an organization does not understand how to manage and control slow projects internally, it will invariably struggle when it out-sources slow projects.



## 2. Customer Myth

1. Myth: A general statement of objectives is sufficient to begin writing programs - we can fill in the details later.

Reality: > A comprehensive & stable statement of requirements is not always possible.

- > How can you expect better result by simply providing general statements alone.
- > There should be continuous communication b/w customer & developer.

2. Myth: Project requirements continually change, but change can be easily accommodated because software is flexible.

Reality: > Cost will increase at final stage.

- > It is true that s/w requirements do change, but the impact of change varies with the time.

## 3. Practitioner's Myth:

1. Myth: Once we write the program and get it work, our job is done.

Reality: > The sooner you begin writing code, the longer it'll take you to get done.

- > 60% to 80% of the effort spent on s/w development.
- > Rework after delivery was done for the customer for the first time.

2. Myth: Until I get the program "running" I have no way of accessing its quality.

Reality: >

- > One of the most effective s/w quality assurance mechanisms can be applied from the inception of the project - the formal technical reviews.



3. Myth: The only deliverable work product for a successful project is the working program.

Reality: > A working program is only one part of a slw configuration that includes many elements.

4. Myth: Software engineering will make us create voluminous and unnecessary documentation and will invariably slow us down.

Reality: > slw engineering is not about creating documents. It is about improving the quality. And reduced rework results in faster delivery times.

### 5) Capability Maturity Model Integration (CMMI)

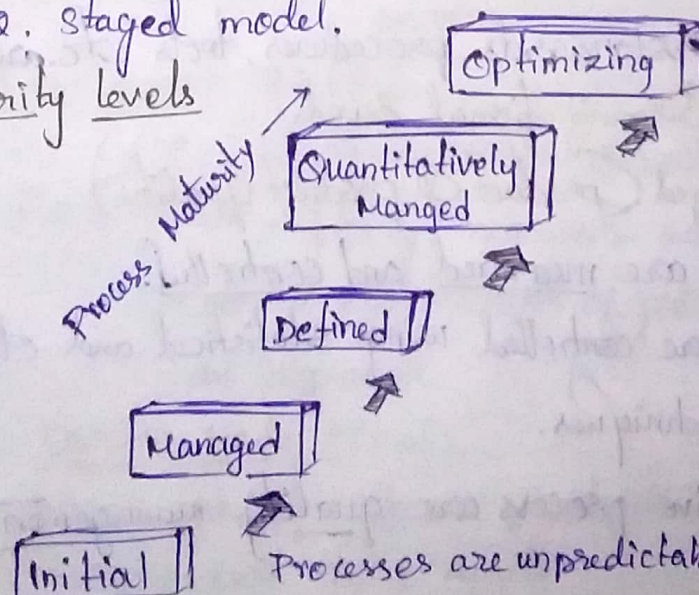
> CMMI defines each process area in terms of "specific goals" and the "specific practices" required to achieve these goals.

> Bench-mark for measuring the maturity of an organization software process.

The CMMI represents a process meta-model in two different ways:

1. Continuous model
2. Staged model.

Maturity levels





## Capability levels:

### Level 1: Incomplete / Initial (Few people)

- > It deals with performed process.
- > Process are unpredictable and poorly controlled.
- > The process performance may not be stable and may not meet specific objectives such as quality, cost and schedule.

### Level 2: Repeatable (Project Management)

- > All of the specific goals of the process area have been satisfied.
- > Software planning.
- > Processes are planned, skilled people, documented, performed, monitored, and controlled at the project level.
- > The managed process comes closer to achieve objectives of cost, quality, schedule.

### Level 3: Defined (definition of process)

- > process definition, training program, peer review.
- > processes are well characterized and understood.

Processes, standards, procedures, tools etc. are defined at the Organizational level.

### Level 4: Managed (product & product quality)

- > processes are measured and controlled.
- > Processes are controlled using statistical and other quantitative techniques.
- > Quantitative process are quality management.



Level 5: Optimized (Continuous process improvement)

- > Process area is adapted and optimized using quantitative means to meet changing customer needs and to continually improve the efficacy of the process area under consideration.

## 6) Personal And Team process model

Personal software process shows engineers how to

- manage the quality of their projects
- make commitments they can meet
- improve estimating and planning
- reduce defects in their products

Framework Activities

1) Planning - Develops size and resource estimation based on requirements.

2) High level Design - external specification of all components

3) High level Design review - Formal verification to uncover errors.

4) Development - Metrics are maintained for all important tasks & work results.

5) Postmortem - Using measures and matrices collected effectiveness of process is determined and improved.

Team process model (TPM)

- It
- ensures quality software products
  - create secure sw products
  - Improve process management in an organization.



## Framework Activities

1. Launch high level design
2. Implementation
3. Integration
4. Test & 5) Postmortem.

## SAQs

1) Software : Software is a collection of computer programs, procedures, rules and associated documentation and data which are collected for specific purpose.

(or)

Software is a set of instructions, programs and data used to operate computers and execute specific tasks.

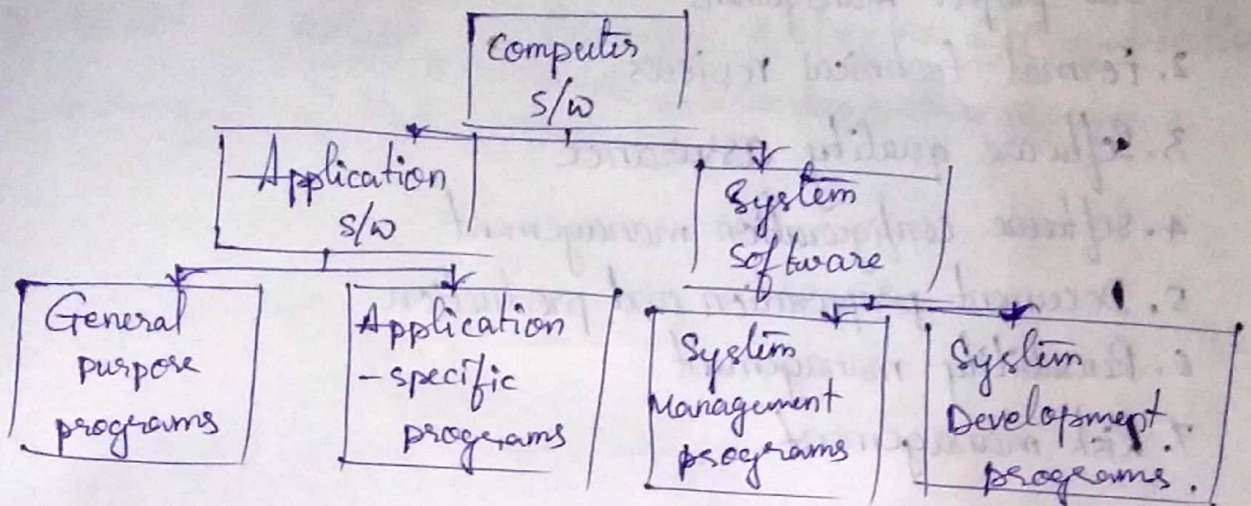
2) Software Engineering: Software Engineering is a detailed study of engineering to the design, development and maintenance of software.

(or)

Software engineering is a process of analysing user requirements and then designing, building and testing software applications which will satisfy those requirements.

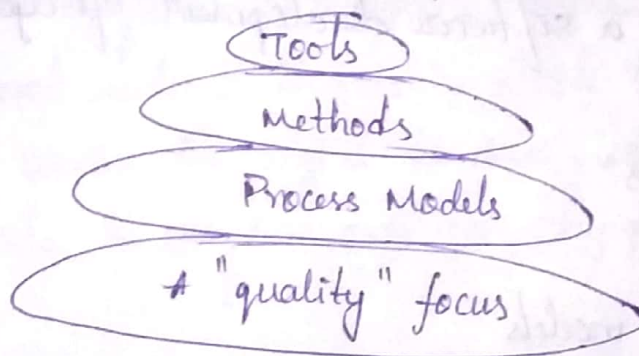


### 3) Categories of Computer Software



4) 5) 6) → Lag ④

7) → A Layered Technology



→ A quality focus: Any engineering approach must rest on an organizational quality for continuous process improvement culture.

Process Models: It defines a framework that must be established for effective delivery of s/w engineering technology.

Methods: s/w methods provide the technical how-to to building software.

Tools: Tools provide automated and semi-automated support for the process and methods.



## 8) Umbrella Activities

1. S/w project management
2. Formal technical reviews
3. Software quality assurance
4. Software configuration management
5. Document preparation and production
6. Reusability management
7. Risk management.

## 9) Process Pattern

Process patterns can be defined as the set of activities, actions, work tasks or work products and similar related behavior followed in a software development life cycle.

## 10) Types of S/w models

- > Waterfall model
- > Incremental process models
- > Evolutionary process models
- > Unified process

11)

Search in google re.



13) Levels in Continuous model in CMMI (it consists of 6 levels).

- 1) Initial 2) <sup>Perform</sup> ~~staged~~ 3) Managed 4) Define 5) Quantitatively managed 6) Optimizing. ~~continuous improvement~~

14)

goto - 11

15) Open-Source Software

Opensource software is a type of computer s/w in which source code is released under a license in which the copyright holder grants users the right to study, change, and distribute the s/w to anyone and for any purpose.

16)

goto - 11

17) Legacy Software: Legacy software is an old and outdated program that is still used to perform a task for a user, even though newer & more efficient options are available.

18) Need of S/W Engineering: It is important because specific s/w is needed in almost every industry, in every business, and for every function. It becomes more important as time goes on... 19) 20) → Lag ⑥.