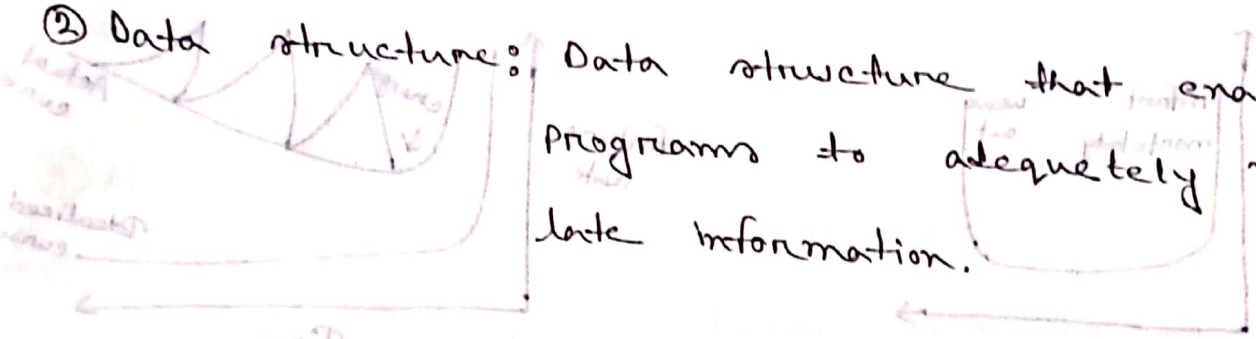
Chapter - 1

(*) Software characteristics (Basic) :

① Instruction: Instruction that when executed provided desired features, functions and performance.

② Data structure: Data structure that enable the programs to adequately manipulate information.



③ Documentation : Documentation that describes the operation and use of programs.

(*) Explain Software characteristics.

⇒

① Software is developed or engineered, it is not manufactured in the classical sense.

manufacture means that to make things, usually on a large scale, with tools and either physical labor or machinary.

- Software development and hardware manufacturing are fundamentally different.
- For hardware, manufacturing phase can introduce quality problems that are non-existent for software.

(b) Software does not wearout

fig: failure curve for hardware

fig: failure curve for software

- It is also called the "bath-tub curve".
- At the beginning of the life of hardware it shows high failure rate as it contains many defects.
- By the time, the manufacturer or designers repair these depects and it becomes idealized or gets into steady state and continues.

- As time passes, failure rate rises again; this may be caused by excessive temperature, dust, improper use and so on, and at one time it become totally unusable. This state is 'wear out' state.

- Software does not wear out but it does deteriorate.

(c) Although the industry is moving toward component based assembly, most software continues to be custom built.

- There components are reusable and in hardware component reuse is common.
- In software world, it is rare.
- A software component must be designed and implemented so that it can be reused in many different programs.

(#) Hardware vs Software.

Hardware	Software
① It is manufactured.	① It is developed or engineered.
② It does wear out.	② It does deteriorate
③ It build using pre-structured electronics components.	③ It build based on specific components such as x86 or x64.
④ Pre-designed architecture so that it is relatively simple.	④ Completely new design so it is relatively complex.
⑤ Hardware failure is random. Hardware does have increasing failure at the last stage.	⑤ Software failure is systematic. Software does not have an increasing failure rate.
⑥ Hardware is physical in nature.	⑥ Software is logical in nature.
⑦ It can not be transferred from one place to another electrically through network.	⑦ It can be transferred.

⑦ If hardware is damaged, it is replaced with new one.

⑧ If software is damaged its backup copy can be reinstalled.

⑨ Hardware is not affected by computer viruses.

⑨ Software is affected by computer viruses.

⑩ Examples: keyboard, Mouse, Monitor, CPU, Printer etc.

⑩ Examples: MS word, Excel, Power point, MySQL etc.

⑪ Legacy software: The definition of legacy software is an old and outdated program that is still performing useful job for community.

An example of legacy software is a factory's computer system running on an old version of windows because there is not a need to invest in the most updated software.

(*) Why must software change?

Ans

(1) The software must be adopted to meet the needs of new computing environments or technology.

(2) The software must be enhanced to implement new business requirements.

(3) The software must be extended to make it interoperable with more modern systems or database.

(4) The software must be re-architected to make it viable within a network environment.

(*) Umbrella activities:

Umbrella activities are applied throughout the software process.

The phases and related steps in software engineering are complimented by a number of umbrella activities:

① Software project tracking and control:

assess progress against the plan and take actions to maintain the schedule.

② Risk management:

assesses risks that may affect the outcome and quality.

③ Software quality assurance:

Defines and conduct activities to ensure quality.

④ Formal technical reviews:

This includes reviewing the techniques that has been used in the project.

⑤ Measurement:

This will include all the measurement of every aspects of the software project.

⑥ Software configuration management:

manage the effects of change throughout the software process.

⑦ Reusability management:

Reuse the functions of software in another software.

⑧ Work product preparation and production:

create work products such as models, documentation logs, forms and lists.

Chapter - 2

Process Model

⑨ Process framework: A framework for the activities, actions, tasks that are required to build high quality software.

- It is a roadmap that helps to create a timely, high quality product or software.
- Also called Software Development Life Cycle. (SDLC)