SOFTWARE ENGINEERING

UNIT – 1

TOPIC – 2 – PART – 1

NATURE OF SOFTWARE AND SOFTWARE APPLICATION DOMAINS

I. Definition

Software is program/ set of programs containing instructions which provide desired functionality. Also comprises of data structures that enable the program to manipulate information

Engineering is the process of designing and building something that serves a particular purpose

Software Engineering is a systematic approach to the development, operation and maintenance of desired software

II. Evolving Role of Software

Software serves as a:

- Product: It delivers the computing potential of a H/W i.e., enables the h/w to deliver the expected functionality. Acts as information transformer
- Vehicle to deliver the product: Helps in creation and control of other programs i.e., it helps other software to do functions and helps as platform. E.g., Operating System

Today, software takes on a dual role. It is a product and, at the same time, the vehicle for delivering a product. As a product, it delivers the computing potential embodied by computer hardware or, more broadly, a network of computers that are accessible by local hardware. Whether it resides within a cellular phone or operates inside a mainframe computer, software is an information transformer—producing, managing, acquiring, modifying, displaying, or transmitting information that can be as simple as a single bit or as complex as a multimedia presentation. As the vehicle used to deliver the product, software acts as the basis for the control of the computer (operating systems), the communication of information (networks), and the creation and control of other programs (software tools and environments). Software delivers the most important product of our time—information. Software transforms personal data (e.g., an individual's financial transactions) so that the data can be more useful in a local context; it manages business information to enhance competitiveness; it provides a gateway to worldwide

information networks (e.g., Internet) and provides the means for acquiring information in all of its forms. The role of computer software has undergone significant change over a time span of little more than 50 years. Dramatic improvements in hardware performance, profound changes in computing architectures, vast increases in memory and storage capacity, and a wide variety of exotic input and output options have all precipitated more sophisticated and complex computer-based systems. The lone programmer of an earlier era has been replaced by a team of software specialists, each focusing on one part of the technology required to deliver a complex application.

III. Why Software Engineering is important?

i. Imposes discipline to work that can become quite chaotic - Lot of steps are involved in the development of a s/w, so if a systematic approach is not taken, it becomes difficult/clumsy

ii. Ensures high quality of software - If a s/w could deliver the features and functionalities required then it is a high-quality software.

iii. Enables us to build complex systems in a timely manner - Whenever we have a huge/complex project, then we need to set proper deadlines and milestones of the time taken in each step, so that in time we can deliver the s/w to the customer.

IV. Difference between Software and Hardware

Software:

- S/W is logical unit
- No spare parts for s/w
- Problem statement may not be complete and clear initially
- Requirements may change with time
- Multiple copies (less cost)
- Idealized and actual graph

Hardware:

- H/W is physical unit
- Spare parts for h/w exist

- Problem statement is clearly mentioned at the beginning of development
- Requirements are fixed before manufacturing
- Multiple copies (more cost)
- Bath tub graph

Bath Tub Curve:

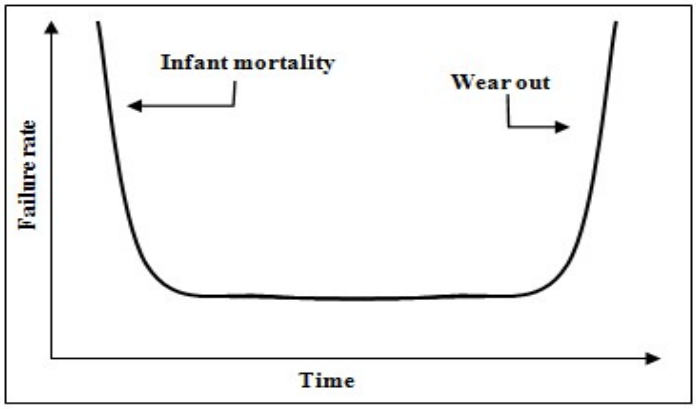

Figure 1: Bath tub Curve- Failure curve for Hardware

- Given the name bath tub because its shape is same as bath tub.
- It explains the reliability of the product i.e. until how many days / time the product works
- We have 3 stages in bath tub curve:
  - Decreasing failure rate or Infant mortality
  - Constant failure rate
  - Increasing failure rate or wear out (H/W effected by the environment factors like dust, temperature, pollution, etc). S/W does not have wear out.
- Decreasing failure rate:
  - As the product in this stage is new, there are very less chances of it to fail. The product still fails because of
    - Manufacturing defect
    - We haven't assembled it properly

- It is a weak part or product.

- Constant failure rate:
  - It is named as constant failure rate because the graph remains in straight line according to the time.
  - Most of the products fail in this stage
  - It is the service life of the product.
- Increasing failure rate:
  - This is the stage where we use the product more than its service period. The suddenly, the product may fail.
  - It is named increasing failure rate as the curve moves upwards as shown according to the time.

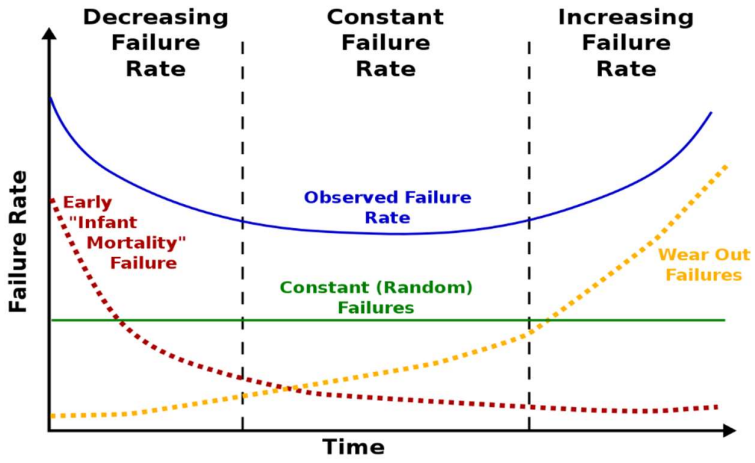

Figure 2: Failure rates in Bath tub Curve

Idealized Curve:

- Since there is no wear out in s/w the curve must undergo on 2 phases i.e., decreasing failure rate and constant failure rate which is called as Idealized curve.
- But in reality, idealized curve is not possible.
- Initial failure happens just like h/w due to undetected defects.

- After correcting the defects, the curve reaches a steady state, but if any change occurs in the s/w then there is spike in the graph.
- Most of the times the failure rate increases when a change effect is requested. The actual curve is higher than the idealized curve.

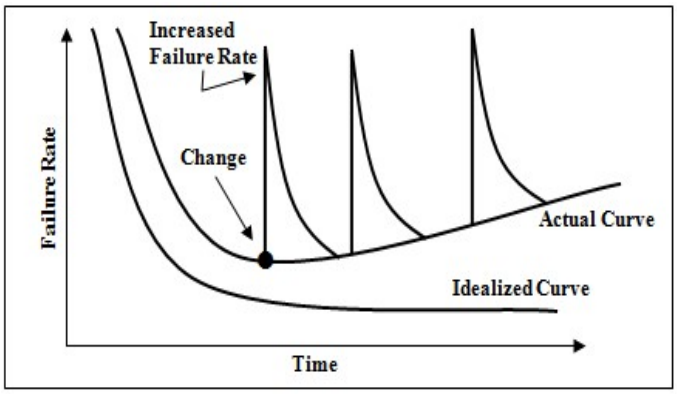

Figure 3: Idealized Curve- Failure curve for Software