UNIT I: Importance of Software Engineering

1.1] Role of Software

Today software takes on a dual role. It is both a product and a vehicle for delivering a product.

As a Product, it delivers the computing potential emboided by a computer hardware.
It is an information transformer - producing, managing, acquiring, modifying, displaying or exmitting information that can be used as simple as a single bit or as complex as multimedia presentation.

As a vehicle, for delivering the product software acts as the basic for the control of the computer, the communication of information & the creation and control of other programs

- Software delivers the most important product of our time - information. It xforms personal data, it manages business information, it provides a gateway to worldwide information networks
- The role of a computer software has changed over a span of more than 50 years. Dramatic improvements in hardware performance, change in computer architecture, increase in main memory & storage capacity, variety of input & output options provided.
- The alone programmer of an earlier era has been replaced by teams of software specialists, each

Scanned by CamScanner

focusing on one part of the technology required to deliver a computer application.

1.2] What is Software?

Software is - instruction that when executed provide desired features, functions and performance.

- Data Structure: They enable the programs to adequately manipulate information
- Documents that describe the operation use of the program.

The gain on understanding of software, it is important to examine characteristics of software which are:

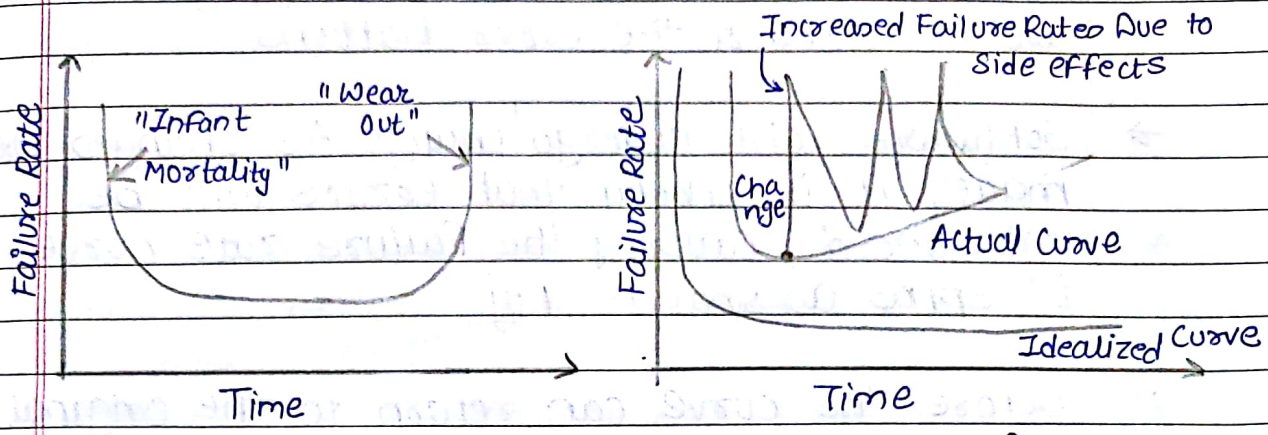
a) Software is developed or engineered, it is not manufactured in the classical sense

→ Although some similarities exists between software development and hardware manufacturing but two activities are fundamentally different
→ In both, high quality is achieved through good design, but the manufacturing phase for hardware can introduce quality problems that are non existent for software
→ Both activities are dependent on people but relationships between people applied & work are different

b)

Scanned by CamScanner

b) Software doesn't "wear out"



a) Failure curve of h/w b) Failure curve for s/w

→ Fig(a) above shows failure rate as a function of time for hardware. The relationships is called as bathtub curve indicates that hardware exhibits relatively high failure rate early in its life

→ Defects are then corrected & failure rate drops to a steady state level

→ As time passed, the failure rate rises again as hardware components suffer from dust vibration, Temprature extremes etc. due to which hardware begins to "wear out"

→ Software is not susceptible to the enviornmental maladies, Therefore the failure rate for software should take to form the idealized curve as shown in fig(b)

Scanned by CamScanner

→ Undiscovered defects will cause high failure rates early in the life of a program. These are corrected & the curve flattens

→ Software will undergo change. AS changes are made, it is likely that errors will be introduced, causing the failure rate curve to spike as shown fig.

→ Before the curve can return to the original steady state failure rate, another change is required, causing curve to spike again.

c) Although the industry is moving towards component based construction, most software continues to be custom built

→ A software component should be designed & implemented so that it can be reused in many different programs.

→ Modern reusable components encapsulates both data & the processing that is applied to data

→ Eg:- Today's user interfaces are built with reusable components that enables the creation of graphics, windows, pull-down menus etc.

Scanned by CamScanner

- Categories of Software

1] System Software

- Is a collection of programs written to service other programs
- Eg:- Compilers, Editors & file management utilities processes complex, but determinate infrastructure
- Other Eg:- OS components, drivers, networking software etc.
- System software area is characterized by heavy interaction with computer hardware, heavy usage of multiple users, resource sharing & process management

2] Application Software

- Application software consists of standlone programs that solve a specific business need

3] Engineering / Scientific Software

- Application range from astronomy to volcanology, from automotive stress analysis to space shuttle orbital dynamics, & from molewlar biology to automated manufacturing
  Eg:- Computer-aided design, system simulation

Scanned by CamScanner

4] Embedded Software

→ Embedded software resides within a product or system and is used to implement and control feature & functions for the end-user & for the system itself.

tg :- Digital function such as fuel control, dashboard displays, breaking system etc.

5] Product Line Software

Designed to produce a specific capability for use by many different customers, product line software can be focus on a limited & estoeric marketplace or address mass consumer markets

Example : Spreadsheets, consumer graphics, multimedia.

6] Web-Application

Web apps can be little more than a set of linked hypertext files that present information using text & limited graphics

Scanned by CamScanner

7.] Artifical Intelligence Software

→ AI software makes use of nonnumerical algorithms to solve complex problem that are not amenable to computation or straightfoorward analysis.
Eg:- Robotics, pattern Recognition, game playing etc

8.] Ubiquitous Computing

→ The challenge for software engineers will be to develop system and application software that will allow small devices, personal computers and enterprise system to communicate across vast network

9.] Open Source

→ A growing trend that results in distribution of source code for system application so that customers can make local modification.

Eg:- DOS, DB etc.

Scanned by CamScanner

SOFTWARE MYTHS

→ It beliefs about software & the process that is used to believe it.

→ They appear to be reasonable statement of facts, they have an intuitive feel, and they are often promulgated by experienced practitioners who "know the score"

① Management Myths

- Managers with software responsibility, like managers in most disciplines, are often under pressure to maintain budgets, keep schedules from slipping & improve

Myth	Reality
• We already have a book that's full of standards & procedures for building software won't that provide my people with everything they need to know	• The book of standards may very well exist, but is it used? Are software practitioner aware of its existence?
• If we get behind schedule we can add more programs & catch up.	• As new people are added, people who are working must spend time educating the new comers, thereby reducing the time spent on productive development effort
• If I decide to outsource the software project to a third party, I can just relax and let the firm build it.	• If an organization does not understand how to manage & control software projects internally, it will invariably struggle when it outsource software projects

Scanned by CamScanner

(2) Customer Myths

A person who requests computer software may be a person at the next desk, a technical group down the hall, the marketing/sales department or an outside company that has requested software under contract.

Myth	Reality
• A general statement of objectives is sufficient to begin writing programs - we can fill in details later	• A poor up-front definition is the major cause of failed software efforts. A formal & detailed description of the information domain, function, behavior, performance, interfaces design constraints
• Project requirements continously change, but change can be easily accomdated because software is flexible	• It is true that software requirement change, but the impact of change varies with the time at which it is introduced change, when required after software is in production can be over an order of magnitude more expensive than the change earlier

Scanned by CamScanner

③ Practitioner's Myths

- Myths that are still believed by software practitioners have been forced by over 50 years of programming culture

Myth	Reality
• Once we write the program and get it to work our job is done	• Someone once said that "the sooner you begin writing code, the longer it'll take you to get done". Industry data indicate that between 60 & 80% of all effort expended on software will be expended after it is delivered to the customer for first time
• Until I get the program "running". I have no way of accessing its quality	• One of the most effective software quality assurance mechanisms can be applied from the inception of a project - the technical review
• The only deliverable work product for a successful project is the working program	• A working program is only one part of a software configuration that include many elements.
• Software engineering will make us create volumin-ous or unneccessary documentation slow us down	• Software engineering is not about creating documents. It is about creating a quality product. Better quality leads to reduced rework.

Scanned by CamScanner

LEGACY SOFTWARE

- The older programs - often referred to as legacy software. Legacy software system were developed decades ago and have been continously modified to meet changes in business requirements & computing platforms

- Many legacy system remains supportive to core business functions and are 'indispensable' to business

- Hence, legacy software is characterized by longevity and business criticality

- Legacy system sometimes have inextensible designs, poor or nonexistent documentation, test cases & results that never archived, a poorly managed change history.

- However, as time passes, legacy system often evolve for one or more of the following reasons:

a) The software must be adapted to meet the needs of new computing environments or technology

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

c) The software must be extended to make it interoperable with other more modern system or software

d) The software must be re-architected to make it viable with in a network environment.

Scanned by CamScanner