UNIT-1 INTRODUCTION TO SYSTEM AND SOFTWARE DEVELOPMENT PROCESS

- Concept of a system
- Basic components of a system
- Information systems categories
- Data Pyramid
- System analyst role and need
- Need of information system development
- Study of different models, Software characteristics, Components,
- Applications, Layered technologies
- Processes, Methods and tools
- Generic view of software engineering
- Classical Systems Development Life Cycle (SDLC) method.

:Concept of a System:

- A System is simply a set of components that interacts to accomplish some purpose.
- In other words, A System is an integrated collection of components which satisfy functions necessary to achieve the system's goals and which have relationships to one another that give cohesion to the system and define structure.
- The word System means different things to different people.
- Systems are all around us, in our everyday life, there are many systems like educational system, computer system, solar system, transportation system, communication system, accounting system, production system and so on.
- A business is also a system. Its components are marketing, manufacturing, sales, research, accounting and personnel. Each of these components is itself a system.
- Every business system depends on a more or less abstract entity called an Information System.

: Basics Components of a System:

Characteristics of the System:

- There are FIVE characteristics of a system are:
  1. Basic components
  2. Interaction and structure
  3. Goal
  4. Behaviour
  5. Life cycle

1. Basic components:

- Every system has a set of interrelated elements or basic components.
- The basic components are simply the various identifiable parts of a system. They are moving parts of a system.
- Thus the basic components may be men, materials, machines, information, concepts or energy.

System	**Basic Components**
Computer	Keyboard, Mouse, CPU, Monitor, ALU, Programs
Education	Students, Teachers, Books, Classes

- If a system is large enough then it is composed of many subsystems.
- Each subsystem is then made up of several smaller subsystems until we reach some parts that individually are not subsystems. These parts are known as Basic Components.

- These components of a system are interrelated and they have certain inter-relationships. For example, in education system, the teacher teaches the student.
- These basic components are that they have certain properties or characteristics.

2. Interaction and Structure:

- An important feature of a system is that the basic components interact among themselves.
- There must be activity or processing procedure between the elements of a system, for, example; in the computer system what is being keyed in gets processed information. Thus in any system, the elements display activity or interaction.
- Interaction alone can establish relationships between these basic components. These relationships that exist among the components and define the boundary between a system and its environment are called its Structure.

3. Goal:

- A system is an organized whole. It has a purpose, goal or objective.
- Without a common objective a system starts moving in all directions and co-ordination among the parts will be lost.
- The goal or purpose merges and integrates the activities of the components. All subsystems and components work more effectively together in the system than if they were acting independently.
- Thus the entire system has to be viewed as a whole rather than just the sum of its parts. This integrating effect is known as 'Synergistic Effect'.

4. Behaviour:

- Behaviour is the way a system reacts to its environment.
- Behaviour is determined by procedures or instructions designed to make sure that components behave in ways that will allow a system to achieve the goals.
- While a procedure describes what should be done, behaviour describes what actually done.
- The reaction is the behaviour and the instructions in the nervous system as to how to react are the procedures.

5. Life Cycle:

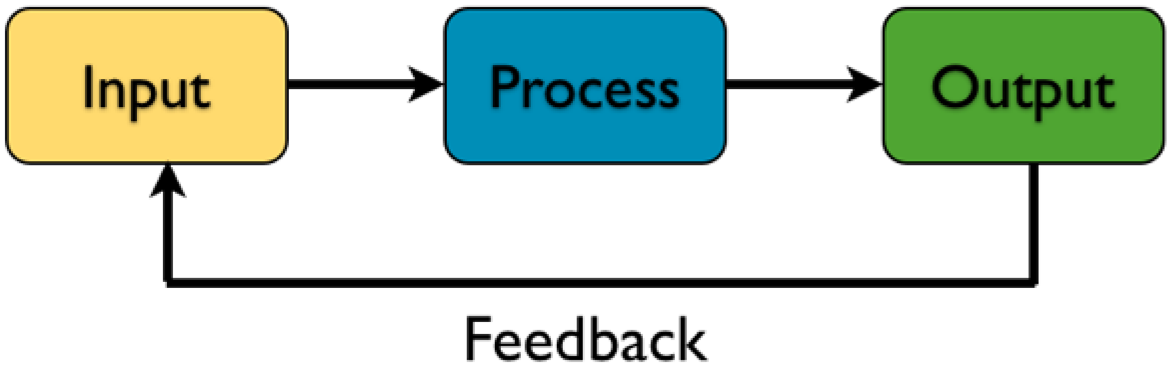
- Every system has birth, life and death.
- Whatever be the system, the life cycle includes evolution, wear, aging, replacement, repair and finally an end to the system's existence.

Elements of a System:

- There are THREE elements of system:

1. Input
2. Processor
3. Output

- **General Model of a system:** A general model of a physical system is made up of system elements like input, processor and output with the system concepts like control and feedback to keep the system in equilibrium is shown in below figure.



[Figure of Elements of System]

- **System Elements:** Input, processor and output are elements common to all systems. They are the elements by which all systems are described. They are set in fixed positions which help the systems analyst to design and work with the system more easily.
  1. **Input:**
     - Input is defined as the energizing or start-up component on which the system operates.
     - The inputs may be raw materials, physical resources, human energy, knowledge or information.
     - Input largely determines the nature of the output. This is more true in information system.
     - Unless the input data are accurate, output information cannot be expected to be accurate.
     - It may be seen that in most cases, output of one system becomes input for the other, e.g. the output of information and decision systems- the decisions – becomes the input to planning and control system.
  2. **Process:**
     - Process is defined as the activity that makes possible the transformation of input to output.
     - Men, machines, functions, operations, organizations and combinations of these may act as processors, e.g. when data are processed through computer, processing involves a set of logical steps.

- These steps are required to be instructed to the computer and this is done by a series of 'instructions' called 'programs'.

3. **Output:**
   - Output is defined as the result of an operation.
   - It is the purpose or objective for which the system is designed.
   - Though output largely depends upon the input, its nature, utility and format may be vastly different from those of input.
   - For example, in information system, input data may be numerical in nature but the output may be pictorial diagrams like pie diagram, bar diagram etc.

- **Example of system elements:**

No.	System	Input	Processor	Output	Feedback
1.	University	Students	Various Department	Changed Students	Alumni Opinions
2.	Management Information System (Computer based)	Information	People, Computer	Decision	Reports about its effectiveness

: Information systems categories:

- There are FIVE types of system:
    1. Conceptual and Physical Systems
    2. Deterministic and Probabilistic Systems
    3. Open and Closed Systems
    4. Natural and Artificial Systems
    5. Man-Machine Systems

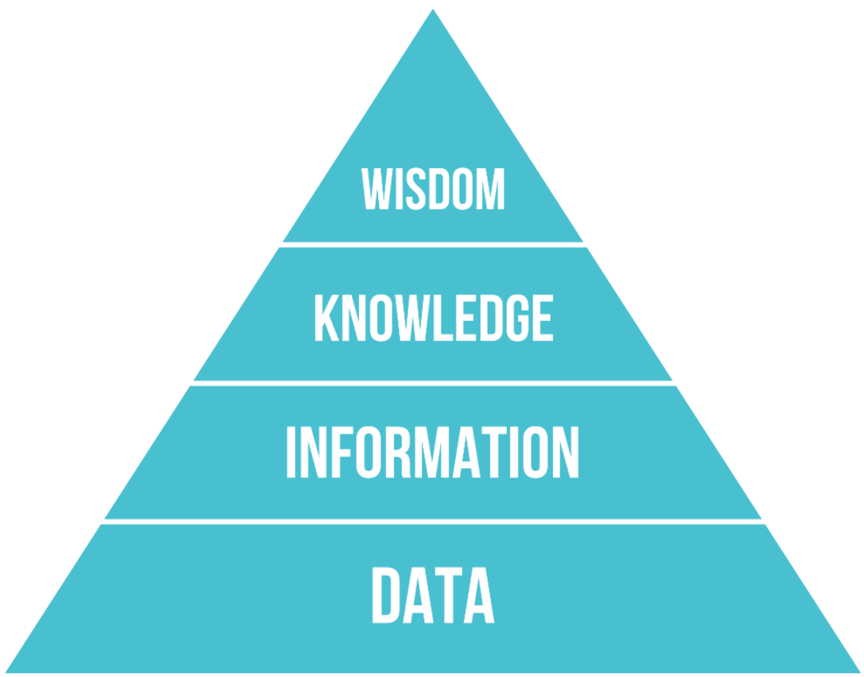
1. **Conceptual and Physical Systems:**
    - System can be abstract (conceptual or analytical) or physical.
    - An abstract or conceptual system is an already is an orderly arrangement of independent ideas.
    - Conceptual systems are concerned with theoretical structures, which may or may not have any counterpart in the real world.
    - In practical management plans, accounting system structures, classifications of policies and procedures are conceptual systems.
    - Thus conceptual systems are systems of explanations or ideas.
    - Physical systems are generally concrete operational systems made up of people, materials, machines energy and other physical things.
    - Physical systems can display activity or behaviour. MIS (Management Information System), Compute system, Business organizations are physically systems.
2. **Deterministic and Probabilistic Systems:**
    - System can be classified as deterministic or probabilistic.
    - A deterministic system operates in a predictable manner. If one knows the state of the system at a given point of time then it is a deterministic system if one can predict the next state without error.
    - For example, the economic forecasting. Similarly, in an inventory system, average demand, lead time etc. are probabilistic. In computer system the outputs are deterministic.
3. **Open and Closed Systems:**
    - An open system is one that interacts with its environment.
    - A business organization is an open system because it exchanges men, material, money and information with the environment.
    - An open system does not provide for its own control or modification. It does not supervise itself. It needs to be supervised by people.
    - A closed system is a system which is self-contained. Is does not exchange material, information or energy with its environment.
    - In organizations and in an information system, we come across systems that are relatively closed.

- A computer system is relatively closed system. A relatively closed system is one that has only controlled and well defined inputs and outputs. It is not to subject to disturbances from outside the system.

4. **Natural and Artificial Systems:**
   - Solar system, water system and human being as a system are examples of natural systems.
   - A business organization, computer system, air conditioning system, social system, economic system, and management information systems are examples of artificial systems.
5. **Man-Machine Systems:**
   - Normally, most of the artificial systems are man-machine systems.
   - A motor car is a machine system. But a motor car can't work without a person. Computer system is machine system. Information system is a man-machine, relatively closed and deterministic system.

: Data Pyramid:

- **The Data Pyramid is also known as, also known variously as the DIKW Pyramid or "DIKW Hierarchy" or "Wisdom Hierarchy" or the "Knowledge Hierarchy", the "Information Hierarchy", and the "Knowledge Pyramid"**, for representing supposed structural and/or functional relationships between **D**ata, **I**nformation, **K**nowledge, and **W**isdom. "Typically information is defined in terms of data, knowledge in terms of information, and wisdom in terms of knowledge".
- The process begins by gathering Data, at the first level, which is then processed to form Information at the second level. When this Information is examined or considered it takes the form of Knowledge, at the third level, and the creation of Knowledge leads to acquiring Wisdom at the fourth or topmost level.



[Figure of Data Pyramid]

- In simple language DIKW is as below:
  - **Data:** raw[basic] facts or material, symbols. data is raw. It simply exists and has no significance beyond its existence (in and of itself). It can exist in any form, usable or not. It does not have meaning of itself. In computer parlance, a spreadsheet generally starts out by holding data.

- **Information:** Information is retrieved from data". Data that are processed to be useful; provides answers to "who", "what", "where", and "when" questions. thereby making the data useful for "decisions and/or action.
- **Knowledge:** Knowledge is typically defined with reference to information. Definitions may refer to information having been processed, organized or structured in some way, or else as being applied or put into action. Application of data and information; answers "how" questions.
- **Wisdom:** Evaluated understanding. Wisdom embodies more of an understanding of fundamental principles embodied within the knowledge that are essentially the basis for the knowledge being what it is. Wisdom is essentially systemic.

* *Data*, forming the base of the hierarchy, can be compiled either manually or through automated systems and value is added to these inputs by interpreting and converting them into a useable and meaningful format termed as *Information*.
* When Information is applied in a particular situation and converted to expertise it comes to be defined as *Knowledge*.
* *Wisdom* is found at the pinnacle of the DIKW hierarchy and differs from data and information, both of which can be harnessed, and also from knowledge which can be shared. Wisdom is more abstract as it is something intrinsic and is typically an accumulation of values, judgments, prior experience or interpretations.
* In organizations, Data and Information are seen as tools to perform tasks and improve operations, while implementing systems and practices creates Knowledge, and the building-up of Knowledge through which transformations and learning takes place generates Wisdom.
* **Example:**
    - Data represents a fact or statement of event without relation to other things.
        - Ex: It is raining.
    - Information embodies the understanding of a relationship of some sort, possibly cause and effect.
        - Ex: The temperature dropped 15 degrees and then it started raining.
    - Knowledge represents a pattern that connects and generally provides a high level of predictability as to what is described or what will happen next.
        - Ex: If the humidity is very high and the temperature drops substantially the atmospheres is often unlikely to be able to hold the moisture so it rains.

- Wisdom embodies more of an understanding of fundamental principles embodied within the knowledge that are essentially the basis for the knowledge being what it is. Wisdom is essentially systemic.
  - Ex: It rains because it rains. And this encompasses an understanding of all the interactions that happen between raining, evaporation, air currents, temperature gradients, changes, and raining.

• :System analyst role and need:

- Individuals who perform the systems investigation as different from those but involved in the detailed computer programming are called "Systems Analyst".
- A systems analyst is like an architect and his work assumes greater importance because he has to design a system for the future.
- Thus a systems analyst designs information systems which meet organizational objective, promote integration of activities, and facilitate control and which are flexible and robust.
- Data processing technology mainly involves systems analysis techniques. That is, systems analysis is the brain for data processing.
- In final systems analysis, it is the systems analyst who determines what data should be processed and how, when and where. So, a system analyst' job consists of:
    1. Gathering facts about existing information system.
    2. Analyzing the basic methods and procedures of current information system
    3. Determining information needs.
    4. Modifying, redesigning and integrating the existing procedures in the new system specifications to provide the needed information.
- The systems analyst is more like a manager who,
    1. Determines the design of the overall system and
    2. Obtains the necessary technical help from programmers, from specialists and equipment engineers
    3. Follows the system through design, implementation, follow-up and re-evaluation.

Role of Systems Analyst

- Following are the different roles of Systems Analyst:

1. **Systems Analyst- an agent of change:**
    - A systems analyst works towards the future. Change is the only thing which is permanent and the systems analyst has to prepare a vehicle to work in that changing environment.
    - The greatest hurdle for him is that people resist change. To overcome, this he has to secure user acceptance through user participation in the design and implementation of the system.
    - Hence, he is the creator of new environments. For this, he has to be a pesuader as well as controller.
2. **Systems Analyst- a motivator:**
    - Acceptance can't be forced down the throats of system users.
    - Proper identification of right personnel and exacting feeding of right motivating factor can go a long way in making a system successful rather than downright.
    - A good motivator has to be a good psychologist.
3. **Systems Analyst- an organizer:**
    - A system is the systems analysts' conceptual child. So, he has to be clear about all activities of the system.
    - The sequence of activities, their purpose and their consequences must be clear to him. He is responsible for the execution of all activities and events of his own system.
    - He should have professional loyalty. He should have the capacity to probe into any problem and arrive at the real cause.
    - He should not take things at face value. 'Digging' must be his second nature.
4. **Systems Analyst- an architect:**
    - A systems analyst must have a fairly good idea of his final system at the raw material stage itself.
    - He prepares the blue print, modifies, improves and provides aesthetic values to his product. For example, a systems analyst may bring in changed attitudes towards systems.
    - So, he is a simplifier, an artist and a sculptor, all rolled into one.
5. **Systems Analyst- an intelligent salesperson:**
    - System selling is harder than that because the systems analyst has to sell it to a user, who knows the existing system in and out.
    - To sell his system he should be a good communicator and genuinely interested in understating the real needs of the user.
    - Logic is the only tool in his hands succeed.

:Need of Information System Analysis:

- Systems analysis and design, as performed by systems analysts, seeks to understand what humans need to analyze data input or data flow systematically, process or transform data, store data, and out-put information in the context of a particular organization or enterprise.
- By doing thorough analysis, analysts seek to identify and solve the right problems.
- Furthermore, systems analysis and design is used to analyze, design, and implement improvements in the support of users and the functioning of businesses that can be accomplished through the use of computerized information systems.
- Installing a system without proper planning leads to great user dissatisfaction and frequently causes the system to fall into disuse. Systems analysis and design lends structure to the analysis and design of information systems, a costly endeavor that might otherwise have been done in a haphazard way.
- It can be thought of as a series of processes systematically undertaken to improve a business through the use of computerized information systems. Systems analysis and design involves working with current and eventual users of information systems to support them in working with technologies in an organizational setting.
- User involvement throughout the systems project is critical to the successful development of computerized information systems. Systems analysts, whose roles in the organization, are the other essential component in developing useful information systems.
- Users are moving to the forefront as software development teams become more international in their composition.
- This means that there is more emphasis on working with software users; on performing analysis of their business, problems, and objectives; and on communicating the analysis and design of the planned system to all involved.
- New technologies also are driving the need for systems analysis.
- Ajax (Asynchronous JavaScript and XML) is not a new programming language, but a technique that uses existing languages to make Web pages function more like a traditional desktop application program. Building and redesigning Web pages that utilize Ajax technologies will be a task facing analysts.
- New programming languages, such as the open source Web framework, Ruby on Rails, which is a combination programming language and code generator for creating Web applications, will require more analysis.

What is Software? And Software Characteristics

Software Definition:

Software is:

(1) Instructions (computer programs) that when executed provide desired features, function, and performance;

(2) Data structures that enable the programs to effectively manipulate information

(3) Descriptive information in both hard copy and virtual forms that describes the operation and use of the programs

Software Characteristics: There are THREE software characteristics:

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

Although some similarities exist between software development and hardware manufacturing, the two activities are fundamentally different.

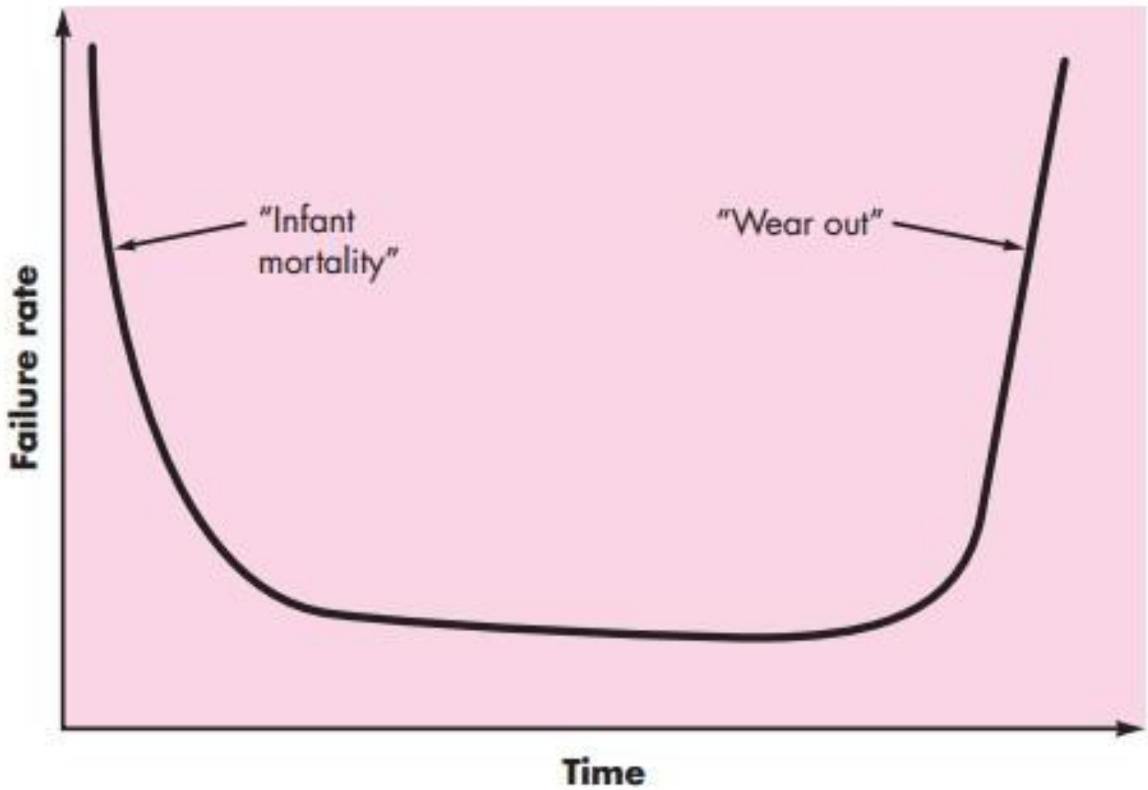
In both activities, high quality is achieved through good design, but the manufacturing phase for hardware can introduce quality problems that are easily corrected for software

Software costs are concentrated in engineering. This means that software projects cannot be managed as if they were manufacturing projects.

2. Software doesn't "wear out."

Figure –A depicts failure rate as a function of time for hardware. The relationship, often called the "bathtub curve," indicates that hardware displays relatively high failure rates early in its life.

As time passes, however, the failure rate rises again as hardware components suffer from the cumulative effects of dust, vibration, abuse, temperature extremes, and many other environmental problems, i.e. the hardware begins to wear out.



[Figure-A- Failure curve for hardware]

In theory, therefore, the failure rate curve for software should take the form of the **"idealized curve"** shown in **Figure-B**.

Undiscovered defects will cause high failure rates early in the life of a program.

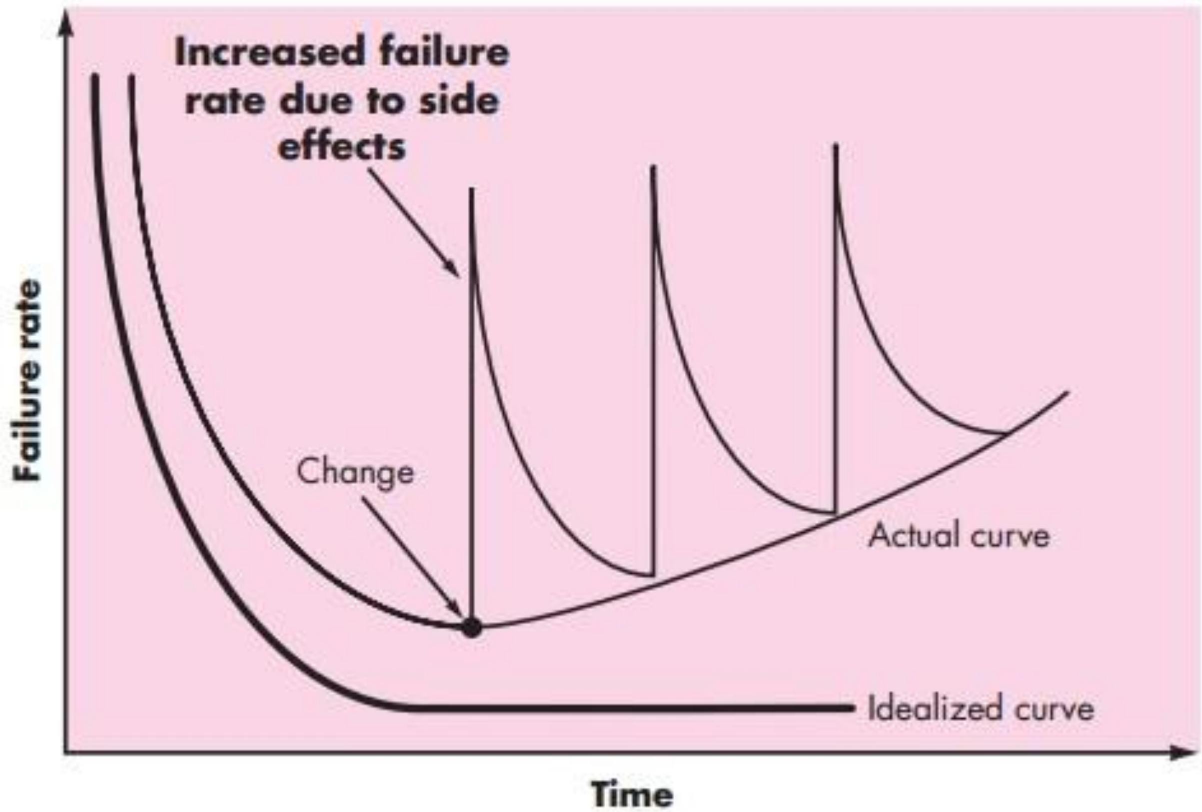
However, these are corrected and the curve flattens as shown.

The idealized curve is a gross oversimplification of actual failure models for software.

However, it is clear—software doesn't wear out.

This seeming contradiction can best be explained by considering the actual curve in Figure-B.

During its life, 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 in the "actual curve"



[Figure-B- Failure curve for Software]

3. Although the industry is moving toward component-based construction, most software continues to be custom built.

A software component should be designed and implemented so that it can be reused in many different programs. Modern reusable components encapsulate both data and the processing that is applied to the data, enabling the software engineer to create new applications from reusable parts.

For example, today's interactive user interfaces are built with reusable components that enable the creation of graphics windows, pull-down menus, and a wide variety of interaction mechanisms.

Software Application

There are **seven** broad categories of computer software present continuing challenges for software engineers:

1) System software—

A collection of programs written to service other pro-grams.

Some system software (e.g., compilers, editors, and file management utilities) processes complex, but determinate, information structures. Other systems applications (e.g., operating system components, drivers, networking software, telecommunications processors) process largely indeterminate data.

2) Application software—

Stand-alone programs that solve a specific business need.

Applications in this area process business or technical data in a way that facilitates business operations or management/technical decision making.

3) Engineering/scientific software—

It has been characterized by "number crunching" algorithms. Applications range from astronomy to volcanology, from automotive stress analysis to space shuttle orbital dynamics, and from molecular biology to automated manufacturing

4) Embedded software—

It resides within a product or system and is used to implement and control features and functions for the end user and for the system itself. Embedded software can perform limited and esoteric functions (e.g., key pad control for a microwave oven)

5) Product-line software—

It is designed to provide a specific capability for use by many different customers. e.g., inventory control products, word processing, spreadsheets, computer graphics, multimedia, entertainment, database management, and personal and business financial applications

6) Web applications—

It is called "WebApps," this network-centric software category spans a wide array of applications. In their simplest form, WebApps can be little more than a set of linked hypertext files that present information using text and limited graphics.

7) Artificial intelligence software—

It makes use of nonnumerical algorithms to solve complex problems that are not amenable to computation or straight forward analysis. Applications within this area include robotics, expert systems, pattern recognition (image and voice), artificial neural networks, theorem proving, and game playing.

8) Legacy software—

Legacy software systems were developed decades ago and have been continually modified to meet changes in business requirements and computing platforms. The proliferation[creation] of such systems is causing headaches for large organizations who find them costly to maintain and risky to evolve.

Software Engineering Layers

Software Engineering Layers:

Software engineering is a layered technology.

Referring to **Figure-C**, any engineering approach (including software engineering) must rest on an organizational commitment to quality.

Total quality management, Six Sigma, and similar philosophies forward a continuous process improvement culture, and it is this culture that ultimately leads to the development of increasingly more effective approaches to software engineering.

There are **FOUR** Software Engineering Layers are as below:

1) A Quality Focus
2) Process
3) Methods
4) Tools

1) A Quality Focus:

The basic layer that supports software engineering is a quality focus.

2) Process:

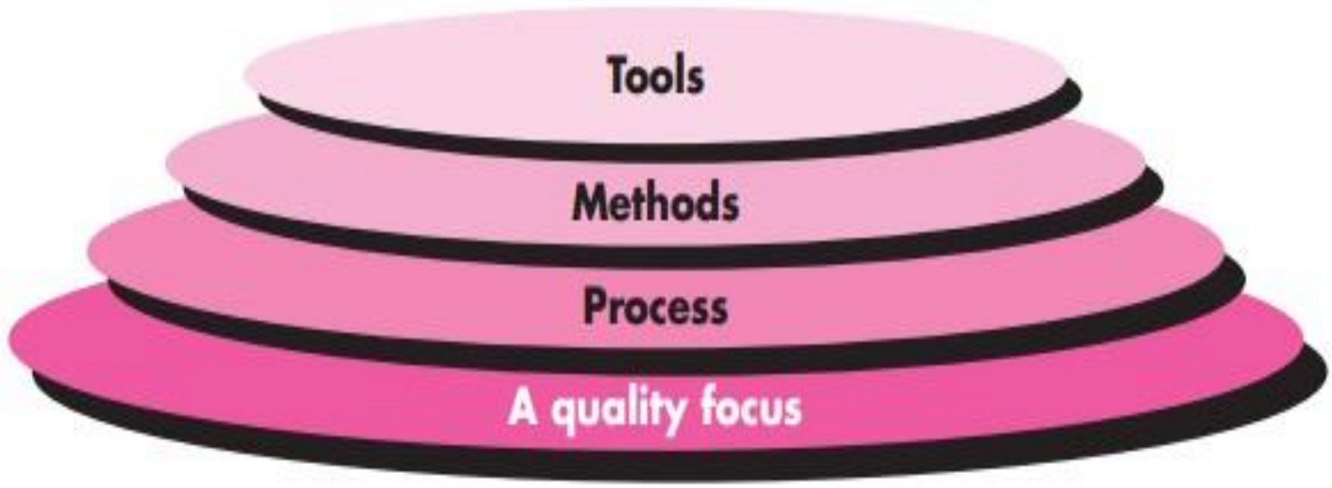
The foundation for software engineering is the process layer.

The software engineering process is the mediator that holds the technology layers together and enables rational and timely development of computer software.

Process defines a framework that must be established for effective delivery of software engineering technology.

The software process forms the basis for management control of software projects and establishes the context in which technical methods are applied, work products (models, documents, data, reports, forms, etc.) are produced, milestones are established, quality is ensured, and change is properly managed.

[Figure-C- Software Layers]



3) Methods

Software engineering methods provide the technical how-to's for building soft-ware.

Methods encompass a broad array of tasks that include communication, requirements analysis, design modeling, program construction, testing, and sup-port.

4) Tools

Software engineering tools provide automated or semi automated support for the process and the methods.

When tools are integrated so that information created by one tool can be used by another, a system for the support of software development, called **computer-aided software engineering,** is established.

SDLC – Classical Waterfall Model

The classical waterfall model is elegant and intuitive. However, it is not a practical model because it cannot be used in actual software development projects. Thus, this model can be considered to be a theoretical way of developing software. The classical waterfall model divides the life cycle into the following phases.

- Feasibility Study
- Requirements Analysis and Specification
- Design
- Coding and Unit Testing
- Integration and System Testing
- Maintenance