Parul® University

Basic Software Engineering

Prof. Vibhuti Patel, Assistant Professor
Parul Institute of Computer Application

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER - 1

Introduction

What is System?

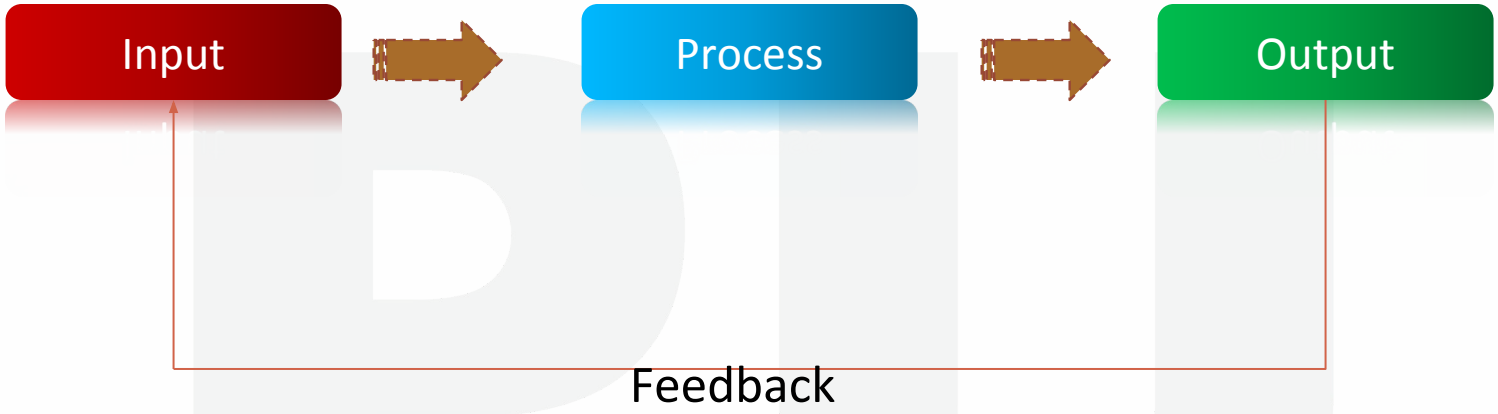
- System is a collection of parts that works together to achieve a specific goal.

Examples:

- Computer System
- Solar System
- Operating System
- Traffic Control System
- Transportation System
- Information System

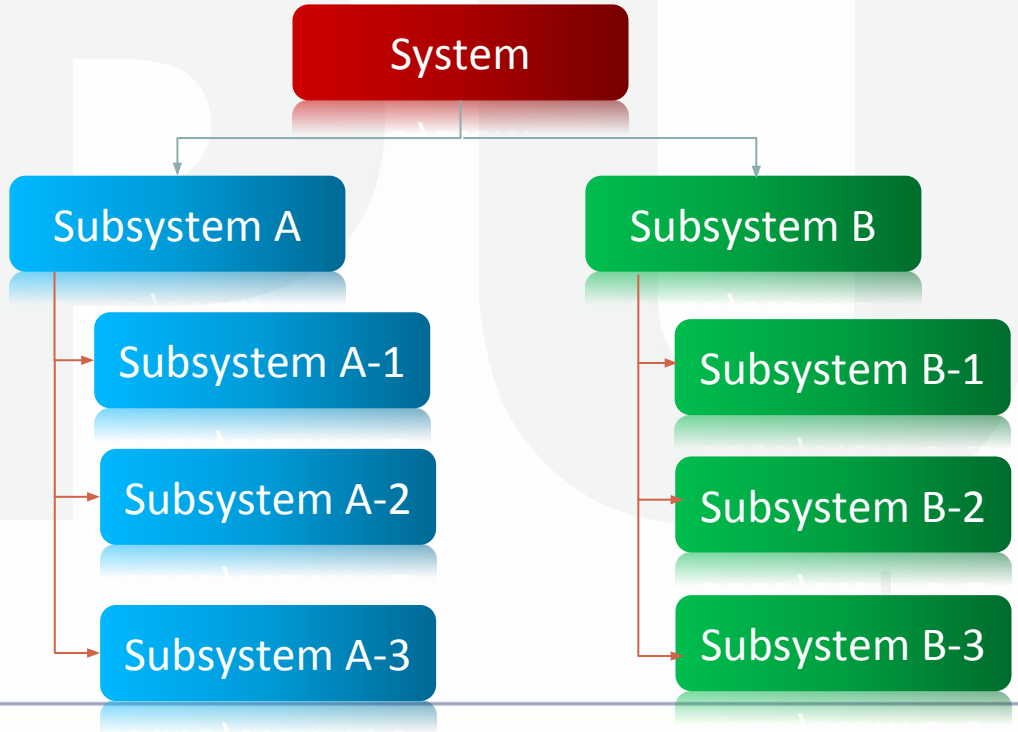
- System is a functional unit, which involves set of procedures/functions to produce certain outputs by processing data/information given as input.

System Elements



What is Subsystem?

- System within a system is called as a subsystem.
- System can be composed of subsystems.



Subsystem Example

- Automobile is a system which is composed of following subsystems:
  - Engine System
  - Body System
  - Frame System

Bad System

- Following are the reasons for Bad System.
  - Failed to meet requirements
  - Poor performance
  - Poor reliability
  - Lack of consumption (usability)
  - Example difficulties:
    - Not to schedule
    - Not to budget
    - Runaway = 100% over budget or schedule

Reasons for failure

- **Complexity**
  - Changes in requirements
  - Poor estimation
  - Poor management
  - Use of new technology
  - Handle complexity by:

    For Example:
    - Proper partitioning of problem
    - Well organized interaction of parts
    - Ensure goal / task achievement

System Concepts

Decomposition: A process of **breaking down** a system into smaller components is known as decomposition.

It allows the systems analyst to:

- Divide a system into small & manageable subsystems
- Focus on one area at a time
- Concentrate on component pertinent (relevant or applicable to a particular matter) to one group of users
- Build different components at independent time

System Concepts

Modularity: A process of **dividing** a system into **modules** of a relatively uniform size is called as a modularity.

- Modules simplify the system design

Coupling: It measures dependency among subsystems.

- There are two types of coupling:
  1. **High Coupling:** Changes to one subsystem will have high impact on the other subsystem.
  2. **Low Coupling:** Changes in one subsystem does not affect any other subsystem.

System Concepts

Cohesion: It measures dependency between classes.

- There are two types of cohesion:
  1. **High Cohesion:** Classes in the subsystem perform similar tasks & are related to each other via many associations.
  2. **Low Cohesion:** Lots of miscellaneous and auxiliary classes, almost no associations.

Stakeholders

Stakeholders are players in the system game.

- Stakeholder term is used to refer to any individual person or group who will be affected by the system, directly or indirectly.
- Stakeholder include end users who interact with a system & everyone else in an organization that may be affected by its installation.
- Stakeholder is any person who has an interest in an existing or new information system.
- Stakeholders can be technical or nontechnical workers.

Stakeholder Classifications

Example:

- For information systems, the stakeholders can be classified as:
  - IS (Information System) Manager: He /she manage secure & effective operations of all computer systems.
  - Systems Analyst: Person who did analysis & design techniques.
  - Programmers: Known as developer who develops software's.
  - End User: Users of the system.
  - Supporting End User: Provides essential supports to the users of the system.
  - Business Manager: Person who manages the business affairs.
  - Other IS mangers/Technician

IS Manager in Systems Development

- A manager of an IS (Information System) department have a direct role in the system development process if the organization is small.

- IS managers are more involved in allocating resources to and overseeing approved system development projects rather than in the actual project development process.

System Analyst

- System Analyst is a key individual in the system development process.
- System Analyst studies the problems and needs of an organization to determine how people, data, processes, communications & information technology can best accomplish improvements for a business.
- Organizational role most responsible for the analysis and design of information systems.

Skills of a Successful System Analyst

- **Analytical Skill:**
  - Understanding of organizations / business
  - Problem solving skills
  - System thinking:
    - Ability to see organizations and information systems as system
- **Technical Skill:**
  - Understanding of potential and limitations of technology
- **Managerial Skill:**
  - Ability to manage projects, resources, risk & change

Skills of a Successful System Analyst

- **Interpersonal Skill:**
  - Effective writing & oral communication skills
  - Help you in work with end user as well as other system analysts and programmers

System Analyst is responsible for

- Efficient capture of data from business source
- Flow of that data to computer
- Processing & storage of that data by the computer
- Flow of useful & timely information back to the business and its people.

Variations on System Analyst Title

- **Business Analyst** is a system analyst that specializes in business problem analysis & technology independent requirements analysis.
- **Programmer / Analyst** includes the responsibilities of both computer programmer and the systems analyst.
- **Others**
  - Systems Consultant
  - Systems Engineer
  - Information Engineer

Programmers

- Programmers convert the specifications or requirements given to them by the analysts into instructions or code that computer can understand.
- **Coding:** Writing a computer program.
- Code generators developed code from specifications, saving an organization time & money.
- Aim of **CASE** (Computer Aided Software Engineering) Tool is to provide a variety of code generators that can automatically produce 90% or more from the system specifications normally given a programmer.

Business Managers

- Another group to system development efforts is business managers such as functional department heads and corporate executives.
- These type of managers are important because they have the power to fund development projects and to allocate resources necessary for projects success.

Other IS Managers/Technicians in System Development

- For Database operations **Database Administrator** is there
- For Network and Telecommunications:
  - **Data Communication Manager**
  - **Voice Communication Manager**
- **Internal Auditors**

Characteristics of Successful Team

- Characteristics are diversity in backgrounds,
  - skills & goals
  - tolerance of diversity, ambiguity & uncertainty
  - complete & clear communication
  - trust
  - mutual respect & putting one's own views second to the team
  - a reward structure that promotes shared accountability & responsibity

System Analysis & Design (SAD)

- **Systems Analysis:** It means understanding & specifying in detail what an information system should do?
- **System Design:** It means specifying in detail how the parts of an information system should be implemented?

What is SAD?

- A complex organizational process whereby computer based information systems are developed & maintained.

Why SAD is important?

- Success of information systems depends on good SAD
- SAD is widely used in industry
- Part of career growth in IT - lots of interesting and well-paying jobs!
- Increasing demand for systems analysis skills

What is Software?

- A software is more than a program code.
- A program is an executable code, which serves some computational purpose.
- A software is considered to be collection of programs (executable programming code), associated libraries & documentations.
- A software which is made for a specific requirement is called software productl.

What is Engineering?

- An engineering is all about developing products or software's using well defined scientific principles & methods.

Software Engineering: It is an engineering approach associated with development of software or product using well defined scientific principles, methods and procedures.

- The outcome of software engineering is a reliable & efficient software product.

Needs of Software Engineering

- **Large Software:** It is easier to build a wall than to a house or building, likewise, as the size of software become large engineering has to step to give it a scientific process.

- **Scalability:** If software process are not based on scientific & engineering concepts, it would be easier to recreate new software than to scale an existing one.

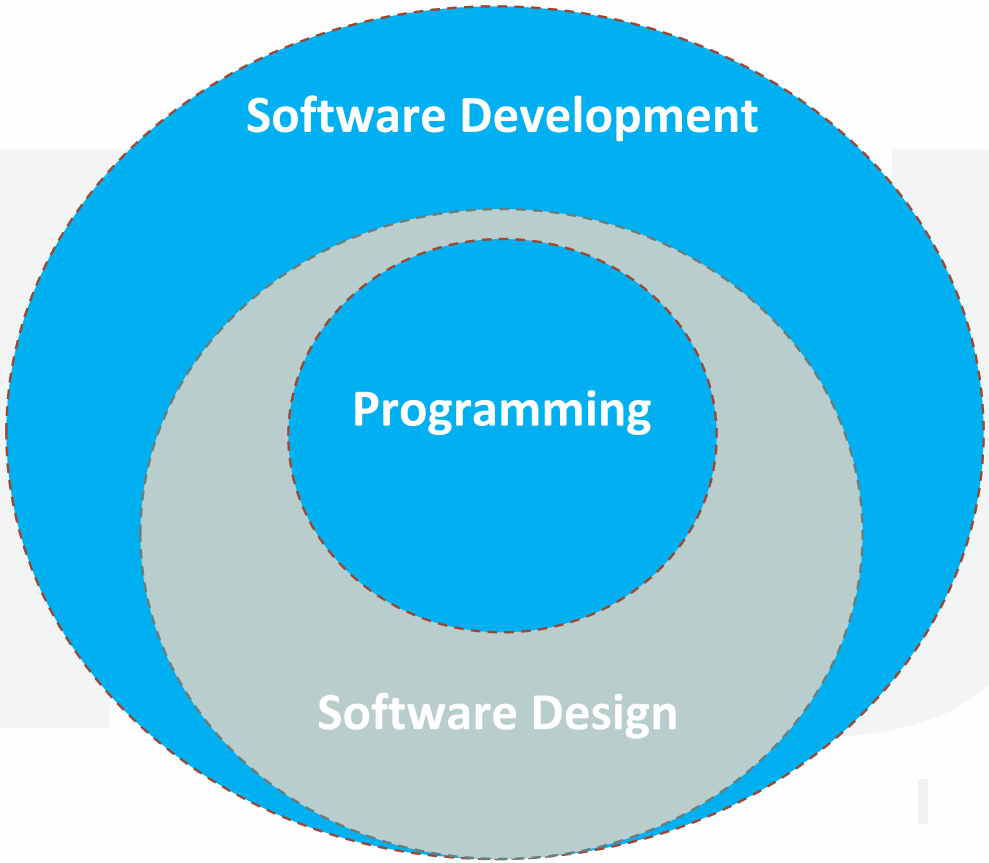
- **Cost:** As hardware industry has shown its skills & large manufacturing has lower down the price of computer & electronic hardware. But cost of software remains high if proper process is not adapted.

Needs of Software Engineering

- **Dynamic Nature:** Always growing & adapting nature of software largely depends upon the environment in which user works. If the nature of software is always changing, new enhancements need to be done in the existing one. This is where software engineering plays a good role.

- **Quality Management:** Better process of software development provides better & quality software product.

Software Model



Software Development Model

- This model is known as **Software Engineering Model** where all engineering aspects that leads to the development of software are applied.
- It includes different requirement gathering & researches which helps the software product to build.

- It consists of
  1. Requirement Gathering
  2. Software Design Model
  3. Programming Model

Software Design Model

- This model is a part of Software Development & it includes:

1. Design
2. Maintenance
3. Programming

Programming Model

- This model is related closely to programming & coding aspect of software development & it includes:

1. Coding
2. Testing
3. Integration

Software Development Life Cycle (SDLC)

- It is a well defined, structured sequence of stages in software engineering to develop the efficient & effective software product.
- **Phases of SDLC:**
  1. Communication
  2. Requirement Gathering
  3. Feasibility Study
  4. System Analysis
  5. Software Design
  6. Coding
  7. Testing
  8. Integration
  9. Implementation
  10. Operation & Maintenance
  11. Disposition

Communication

- This is the first phase where the user initiates the request for a desired software product and starts communication.

- The user communicates with the service provider and tries to negotiate the terms, submits the request to the service providing organization on paper.

Requirement Gathering

- This step onwards the software development team works to carry on the project.
- A team holds discussions with various stakeholders from problem domain & tries to bring out as much information as possible on their requirements.
- The requirements are contemplated and segregated into user requirements, system requirements and functional requirements.

The requirements are collected using a number of practices as given

1. studying the existing or obsolete system and software
2. conducting interviews of users and developers
3. referring to the database
4. collecting answers from the questionnaires.

Requirement Gathering Techniques

- Background reading
- Interviewing
- Observation
- Document Sampling
- Questionnaires

We will discuss all techniques in next unit.

Feasibility Study

- After requirement gathering, the team comes up with a rough plan of software process.
- At this phase the team analyzes if a software can be designed to fulfill all requirements of the user, and if there is any possibility of software being no more useful.
- It is also analyzed if the project is financially, practically, and technologically feasible for the organization to take up.
- There are many algorithms available, which help the developers to conclude the feasibility of a software project.

System Analysis

- At this phase the developers decide a roadmap of their plan and try to bring up the best software model suitable for the project.
- System analysis includes understanding of software product limitations, learning system related problems or changes to be done in existing systems beforehand, identifying and addressing the impact of project on organization and personnel etc.
- The project team analyzes the scope of the project and plans the schedule and resources accordingly.

Software Design

- Next phase is to bring down whole knowledge of requirements and analysis on the desk and design the software product.
- The inputs from users and information gathered in requirement gathering phase are the inputs of this step.
- The output of this step comes in the form of two designs; logical design, and physical design.
- Engineers produce meta-data and data dictionaries, logical diagrams, data-flow diagrams, and in some cases pseudo codes.

Coding

- This phase is also known as programming phase.

- The implementation of software design starts in terms of writing program code in the suitable programming language and developing error-free executable programs efficiently.

Testing

- An estimate says that 50% of whole software development process should be tested. Errors may ruin the software from critical level to its own removal.

- Software testing is done while coding by the developers and thorough testing is conducted by testing experts at various levels of code such as module testing, program testing, product testing, in-house testing, and testing the product at user's end.

- Early discovery of errors and their remedy is the key to reliable software.

Integration

- Software may need to be integrated with the libraries, databases, and other program(s).

- This stage of SDLC is involved in the integration of software with outer world entities.

Implementation

- This means installing the software on user machines. At a time, software needs post installation configurations at user end.

- Software is tested for portability and adaptability and integration related issues are solved during implementation.

Operation & Maintenance

- This phase confirms the software operation in terms of more efficiency and less errors. If required, the users are trained on, or aided with the documentation on how to operate the software and how to keep the software operational.

- The software is maintained timely by updating the code according to the changes taking place in user end environment or technology.

- This phase may face challenges from hidden bugs and real-world unidentified problems.

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in