

AGILE-PARADIGM SHIFT IN SDLC

Session - 2

Software Development Life Cycle



Objectives

- ◆ Explain SDLC
- ◆ Explain evolution of SDLC
- ◆ Identify the need for SDLC
- ◆ Identify and describe the phases of SDLC
- ◆ Describe a case study on SDLC

For Aptech Centre Use Only

Introduction [1-3]

◆ Software Development Life Cycle (SDLC)

- ◆ Is a process applied to the development of a software product.
- ◆ Consists of a sequence of activities or phases that govern the entire life cycle of a software development.
- ◆ Phases may vary from project to project, however, the basic concept of SDLC remains the same.

SDLC process help organizations to plan their software development projects efficiently and allocate the right resources to the project.



Introduction [2-3]

◆ Application of SDLC

- ◆ SDLC is applied to build new software projects and also to enhance existing software.
- ◆ SDLC is applied right from when the project is planned and extends throughout the development phase, until the software is installed or ready for deployment.

For Aptech Centre Use Only

Introduction [3-3]

- ◆ SDLC has to meet a company's business requirements which are as follows:

High-quality Software Systems

- Developers and customers have to strike a balance between desired level of quality and cost.

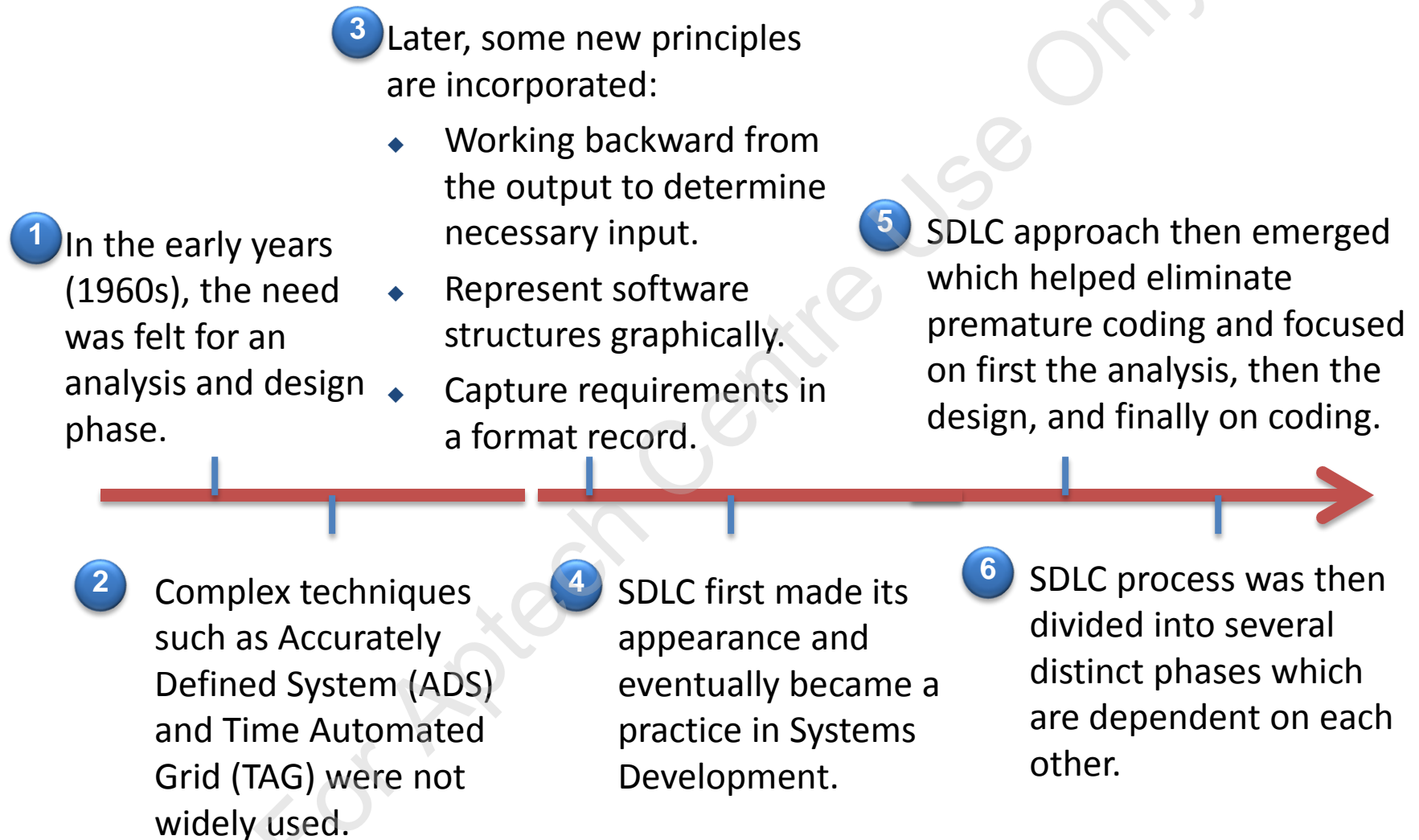
Strong Management Control

- Throughout a project, regular checks are made to see if it is on track. If not, then corrective measures need to be taken.

Increased Productivity

- Businesses and software developers have to choose the approach that yields maximum results for a given cost.

Evolution of SDLC



Need for Software

- ◆ Some of the reasons for need for software are:

There is an increasing need for automation of different everyday services such as processing of bills, personal banking, ticket booking, and so on.

As businesses evolve, their services also need to evolve, which in turn means better and more complex software.

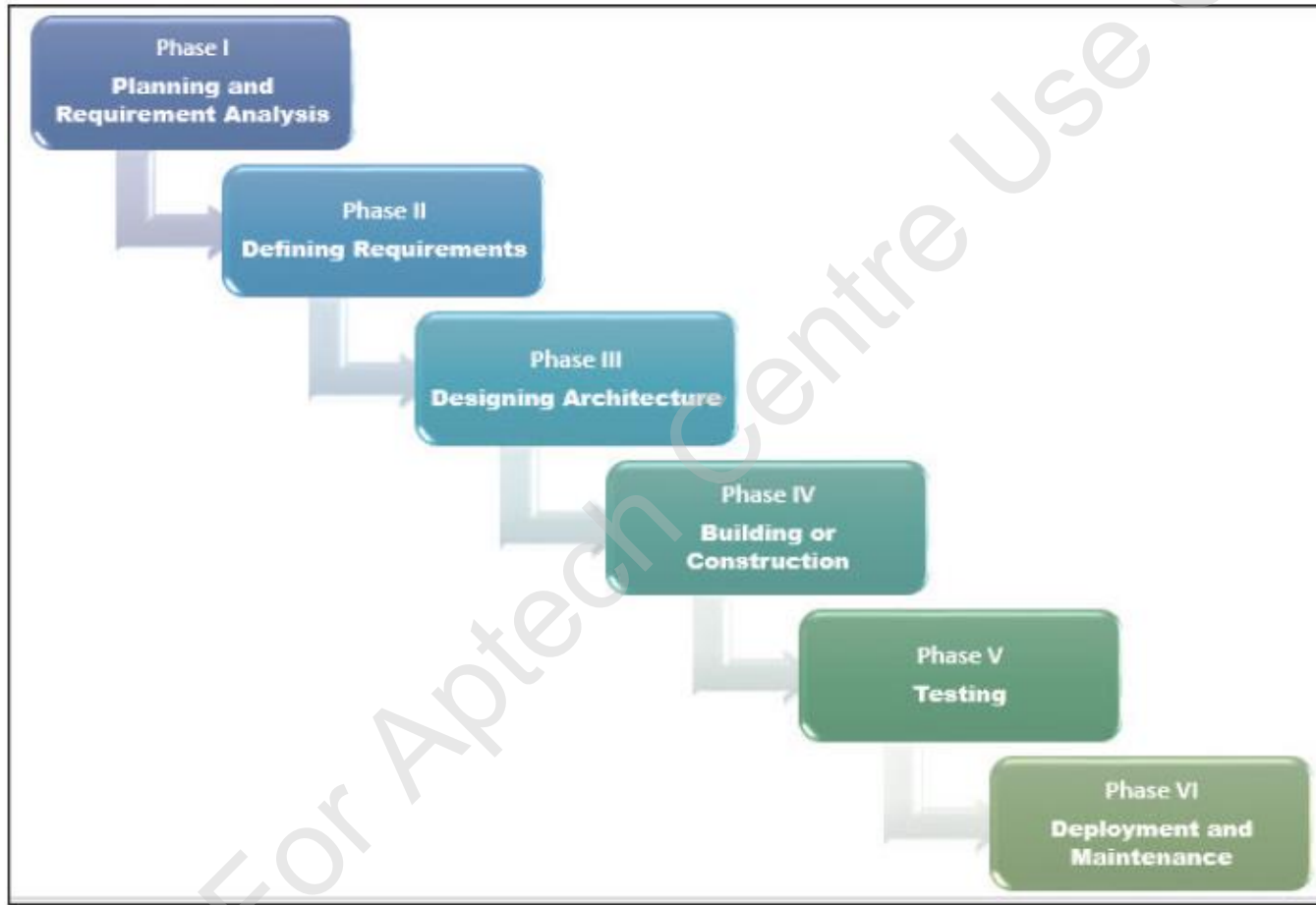
Software processes have to be designed to handle a wide variety of inputs at the interface level. For example, personal details entered while booking tickets.

There should be no errors in a software, especially while using for sensitive purposes, such as updating medical records, no matter how often or number of times it is run.

Integrating software component with other parts of the application should be easy.

SDLC Phases

- ◆ Figure depicts a typical SDLC phases or key activities that have to be part of any software development activity.



Planning and Requirement Analysis [1-4]

- ◆ The SDLC starts with:
 - ◆ **Problem Definition Phase** – In this, client's problem is identified and understood.
 - ◆ **Requirement Gathering** – Involves conducting a preliminary meeting with the client to understand and analyze the client's needs.
- ◆ Figure shows different techniques used in requirement gathering.



Planning and Requirement Analysis [2-4]

Interviewing

- ◆ Is an often-used technique when an analyst tries to get information regarding a problem from managers, department heads, and functional heads.
- ◆ The analyst will ask the customer a set of context free questions which:
 - ◆ Will lead to a basic understanding of their problem and nature of the solution that is desired.
- ◆ The context free questions can be:



Planning and Requirement Analysis [3-4]

Questionnaires

- ◆ Are used when a wide spectrum of responses is required and the consultants need definite answers to certain questions related to the problem area.
- ◆ For instance, getting responses from workers at a factory site is an ideal situation for floating questionnaires as a lot of time can be taken up interviewing such a large group.

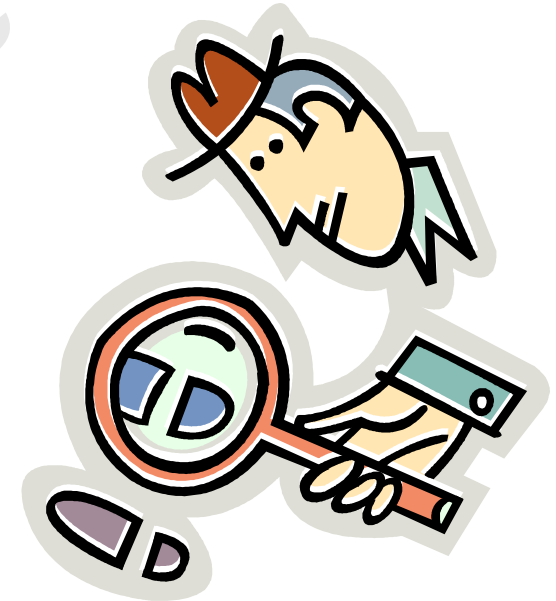
Record Reviews

- ◆ A technique used in all projects.
- ◆ Helps to review existing records that an analyst can understand the data, its movement, and usage within an organization.
- ◆ Cuts down on the time spent in asking people within the organization basic questions that relate to the day to day functioning of a system.

Planning and Requirement Analysis [4-4]

Observation

- ◆ Helps the analyst in detecting problems that exist in the current system.
- ◆ Users:
 - ◆ Are either unaware of the problems or they are hesitant to talk about them to outsiders, thinking that they may be misinterpreted.
 - ◆ May ignore certain minute and delicate details, which may be of extreme importance to the analyst.
- ◆ Such problems may come to light when an analyst observes the way an organization functions.



Defining Requirements [1-7]

- ◆ Having identified the problem, the system development can be started.
- ◆ The process of system development starts with '**System Analysis**' which:
 - ◆ Is the specification of what the system is required to do.
- ◆ System Analysis is conducted with the following objectives in mind:

Identification and analysis of customer needs

System evaluation for feasibility

Performing economic and technical analysis

Allocating functions to manpower, database, hardware, software, and other system elements

Establishing cost and schedule constraints

Creating a system definition that forms a foundation for all subsequent development activities

Defining Requirements [2-7]

- ◆ System Analysis study normally:
 - ◆ Conducts a feasibility study, based on its result.
 - ◆ Performs the translation phase of inputs into formal specified requirements.

Feasibility Study

- ◆ A feasibility study is conducted to find out whether the proposed system will be:

Possible

- To build it with the given technology and resources.

Affordable

- Given the time and cost constraints of the organization.

Acceptable

- For use by the eventual users of the system.

Defining Requirements [3-7]

- ◆ Feasibility study provides answers to troublesome questions:
 - ◆ Can I undertake this project?
 - ◆ What are my market constraints?
 - ◆ What is competition in this field like?
 - ◆ How do I price my product?
 - ◆ What benefits will accrue from this project?
- ◆ A feasibility study is initiated by an organization, which feels there is a need for change in the current system of functioning.
- ◆ Based on the outcome of the feasibility study, the management decides to proceed to the next phase of requirement analysis.



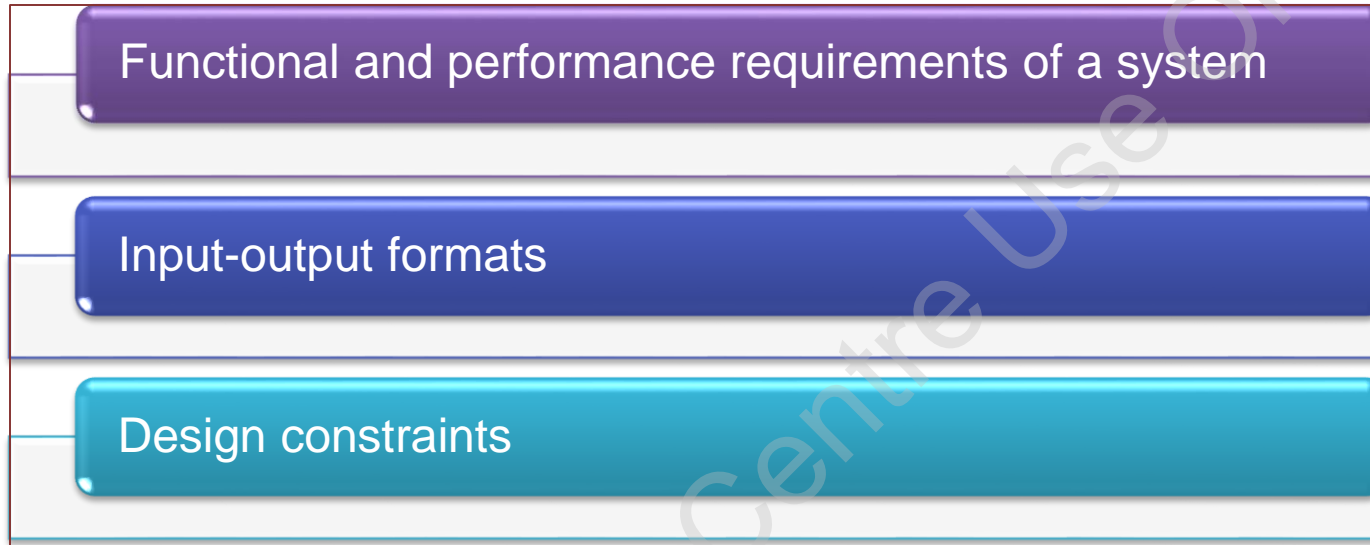
Defining Requirements [4-7]

Translation of Inputs

- ◆ This phase translates these informal inputs from the users into a set of formal specified requirements.
- ◆ The process of requirement analysis helps to bridge the communication gap between the customer and the developer.
- ◆ The technical development team works with the customers and system end-users to identify:
 - ◆ Application domain
 - ◆ Functions and services
 - ◆ Performance capabilities
 - ◆ Hardware constraints
- ◆ The requirements must be specified in the **Requirement Specification Document (RSD)**.

Defining Requirements [5-7]

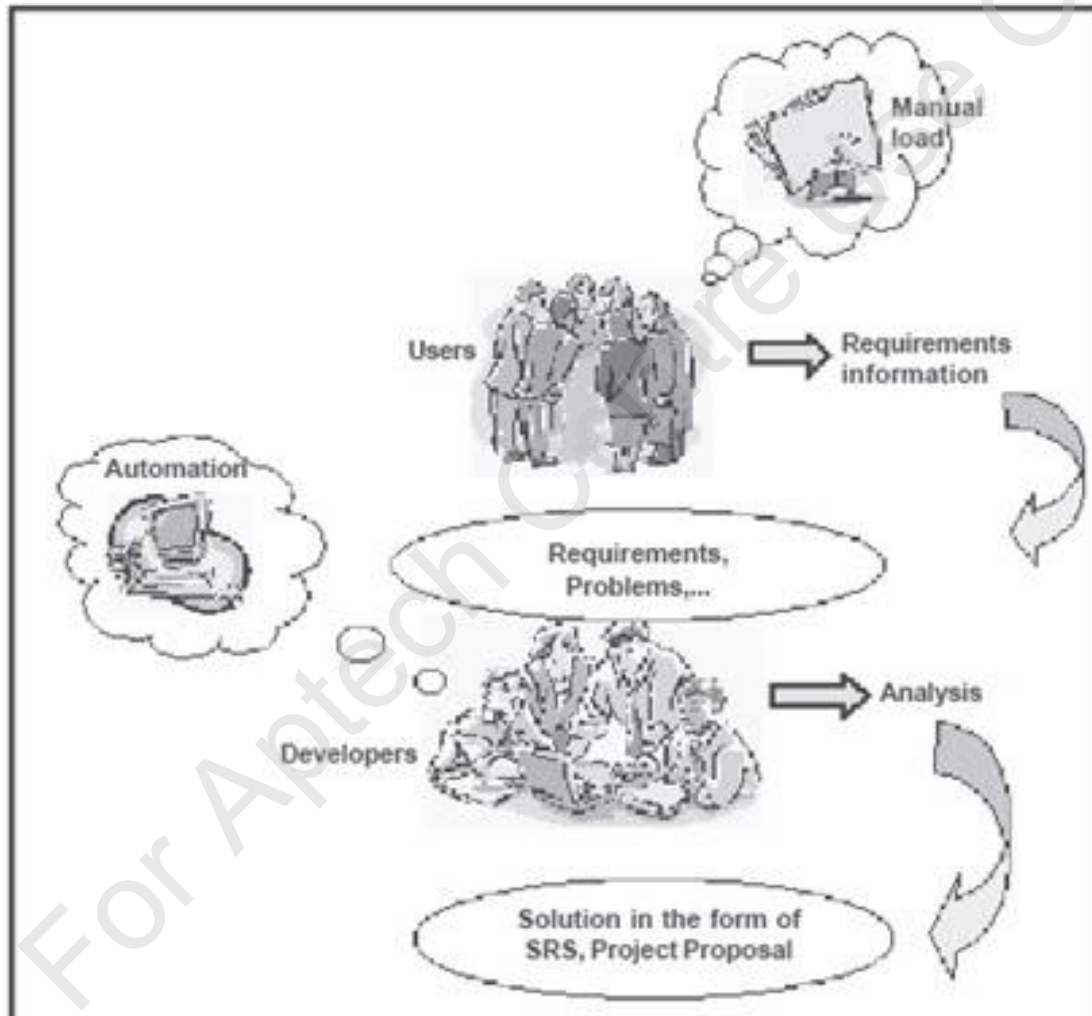
- ◆ The contents of the RSD are:



- ◆ After a system has been analyzed and its requirements determined, they must be placed in the context of the functions that will be ultimately incorporated in the system.

Defining Requirements [6-7]

- ◆ Figure summarizes the process of defining Requirement Analysis.



Defining Requirements [7-7]

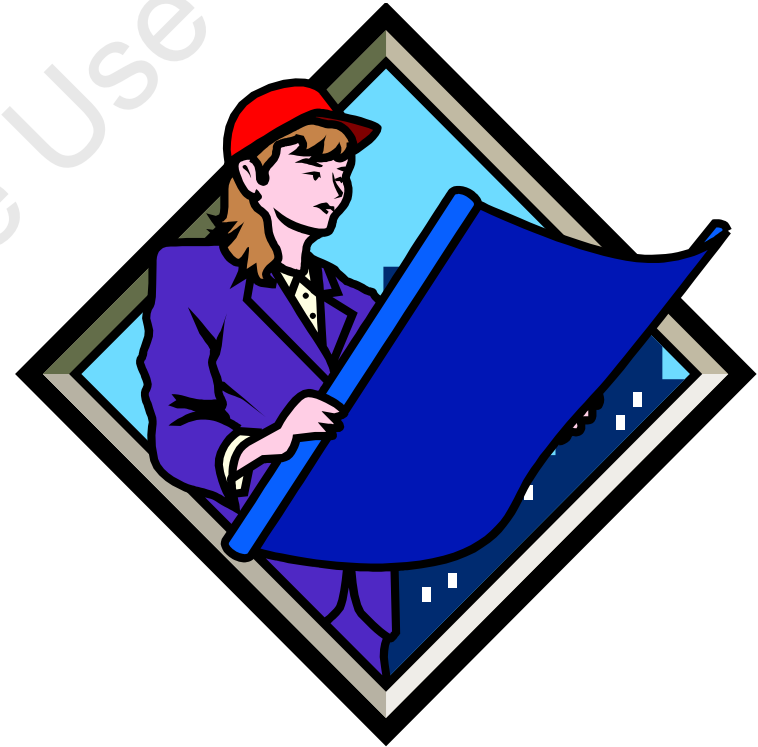
- ◆ **Example: A car is to be developed for the middle-income group of the population.**
 - ◆ Initial study is conducted with the target group of the population.
 - ◆ Requirements state that it must be an affordable vehicle, fuel efficient, easy to run, good braking system for heavy traffic manoeuvres, and compact size for easy parking.
 - ◆ The proposed solution should consider:
 - ◆ **Functions of the car** – power brakes, easy shift floor gear system, power steering
 - ◆ **Performance requirements** – fuel-efficient engine to give 'x' miles per gallon
 - ◆ **Design constraints** – size of the chassis

Based on the given requirements and expected functionality, a suitable car is designed for the middle income population.



Designing Architecture [1-5]

- ◆ One of the most important stages during the SDLC is the design stage.
- ◆ **Design stage:**
 - ◆ Is intricate and complex.
 - ◆ Can be defined as the process of applying various techniques and principles to define a device, process, or system.
 - ◆ Is an iterative process which transforms requirements into an outline for constructing the software.
 - ◆ Is situated at the core of the software engineering process.
 - ◆ Is always applied irrespective of the software process model used.



Designing Architecture [2-5]

- ◆ The different elements are used to provide information necessary to create a design model. These are as follows:
 - ◆ Entity Relationship (ER) diagram
 - ◆ Data Flow Diagram (DFD)
 - ◆ Control Flow diagram
 - ◆ Data dictionary
- ◆ The design model comprises data design, architectural design, interface design, and procedural design.

Designing Architecture [3-5]

Architectural design

- Defines the relationship among the major structural elements of the program.

Interface design

- Describes how software communicates with systems that interoperate with it and with people who use it.

Component-level design

- Transforms structural elements of the program architecture into a procedural description of software components.

Designing Architecture [4-5]

- ◆ The following three characteristics of design have been suggested by McGlaughlin and they serve as a guide for the evaluation of a good design:

The design must implement all the explicit requirements contained in the analysis model.

It must accommodate all the implicit requirements desired by the customer.

The design must be a guide that is easily understood by those who write code and for those who test and subsequently maintain the software.

The design should provide a complete picture of the software, address the data, and functional and behavioral domains from an implementation perspective.

Designing Architecture [5-5]

- ◆ Technical criteria to establish a good design can be summarized as follows:

A design should exhibit a hierarchical organization that makes intelligent use of control among elements of software.

A design should be modular, which means software should be logically partitioned into elements that perform specific functions and sub-functions.

A design should contain both data and procedural abstractions.

A design should lead to interfaces that reduce the complexity of connections between modules and the external environment.

A design should be derived using a repetitive method that is driven by information obtained during software requirements analysis.

Building or Construction

- ◆ This is the phase where the actual coding of a software application takes place, and a product is built from scratch.
- ◆ This is an implementation phase where a number of activities running in parallel have to culminate.
- ◆ Standard activities to be performed during the implementation process:

Creation of an Installation Plan

Implementation of Physical Procedures

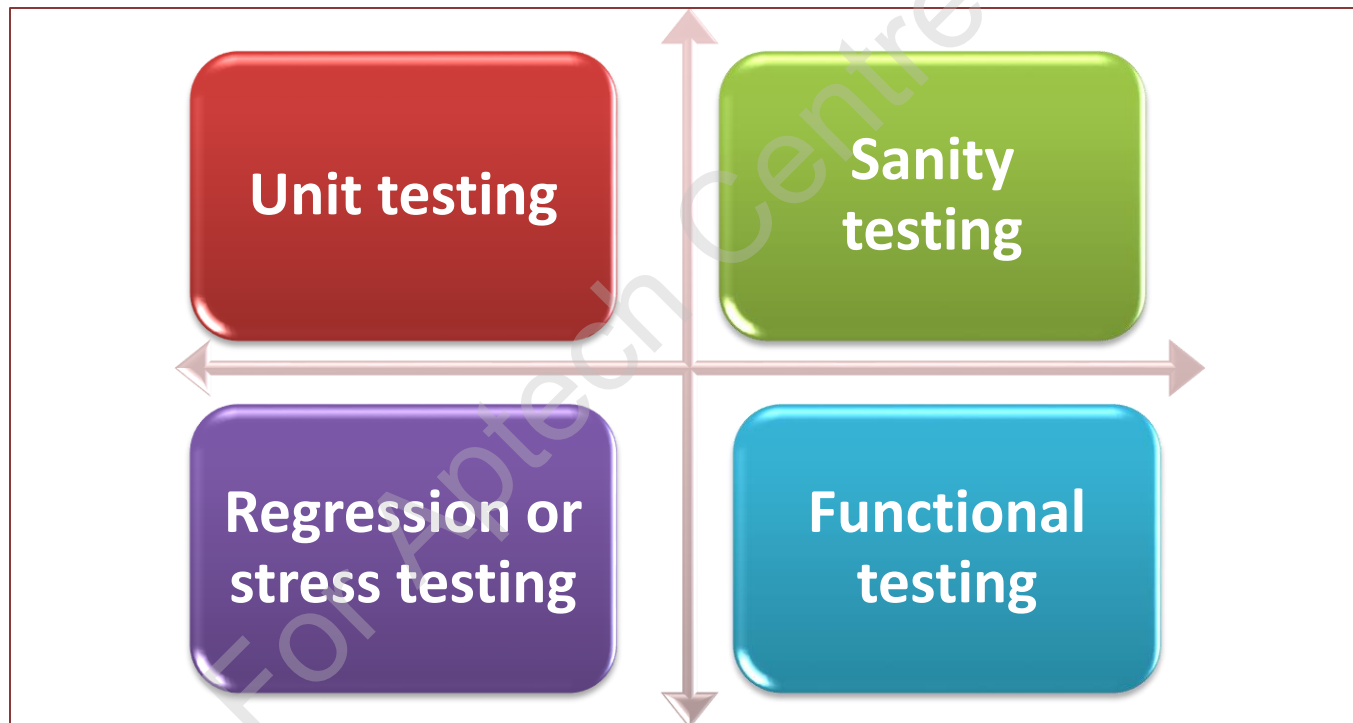
Data Preparation and Conversion

Conducting User Training

Running the System

Testing

- ◆ Testing is very important in SDLC as the quality of the product depends on this phase.
- ◆ The efficiency of the testing phase can make or break a software product.
- ◆ Testing is also called as Quality Assurance (QA).
- ◆ The different types of testing conduct on the software are as follows:



Deployment and Maintenance

- ◆ Maintenance activities involve:
 - ◆ Making enhancements to the software products developed in the earlier stages of the life cycle.
 - ◆ Adapting products to new environments.
 - ◆ Correcting problems.
 - ◆ Making enhancement of the product may in turn involve providing new operational capabilities.
 - ◆ Improving the user screens and modes of interaction.
 - ◆ Improving the performance of the system in general.
- ◆ Adapting the product may require moving the project to a different machine or platform, or accommodating new protocols or even new hardware components such as an additional disk drive.

Case Study [1-2]

- ◆ **ABC Insurance Co. fixes on the need for a customer information system. The company feels this will help them track the customer's policy and profile details. The system will provide better visibility on plans that are currently the best sellers in the market besides giving the marketing team the inputs to build on these plans.**
- ◆ Michael Philips, an industry expert in Software modeling, is hired by the company to help them realize this goal.
- ◆ Michael recognizes from the outset that this particular project would follow the stages of a traditional SDLC.
- ◆ Considering the current level of expertise in the organization, he also proposes that the testing of the application be outsourced to third-party vendors to minimize the turnaround time of the application and provide accurate test case results.

Case Study [2-2]

- ◆ How will the SDLC phases look for this case?
 - ◆ Requirements Gathering and Analysis
 - ◆ Designing Architecture
 - ◆ Building and Testing
 - ◆ Operations and Maintenance



Summary [1-2]

- ◆ Software Development Life Cycle (SDLC) is a process applied to the development of a software product. This process consists of a sequence of activities or phases that govern the entire life cycle of a software development.
- ◆ A typical SDLC consists of the phases namely, Planning and Requirement Analysis, Defining Requirements, Designing Architecture, Building or Construction, Testing, and Deployment and Maintenance.
- ◆ System Analysis Study normally conducts a feasibility study and based on its result, the translation phase of inputs into formal specified requirements is performed.
- ◆ Design can be defined as the process of applying various techniques and principles to define a device, process, or system in sufficient detail to permit its physical realization.

Summary [2-2]

- ◆ Implementation Planning is an important activity of the SDLC. The specific approach adopted for implementation is largely dependent on the size of organization, the nature of applications, and the standard practices that may have been evolved by the project team.
- ◆ Testing is very important in SDLC as the quality of the product depends on this phase. The efficiency of the testing phase can make or break a software product.
- ◆ Maintenance activities involve making enhancements to the software products developed in the earlier stages of the life cycle, adapting products to new environments, and correcting problems.

For Aptech Centre Use Only