# Exploring GDB commands

**Assignment-2**

G R O U P - 7

**MEMBERS:**

1. Dipayan Maity 2021CSB039

2. Aaratrika Banerjee 2021CSB040

3. Pantho Propan Debnath 2021CSB041

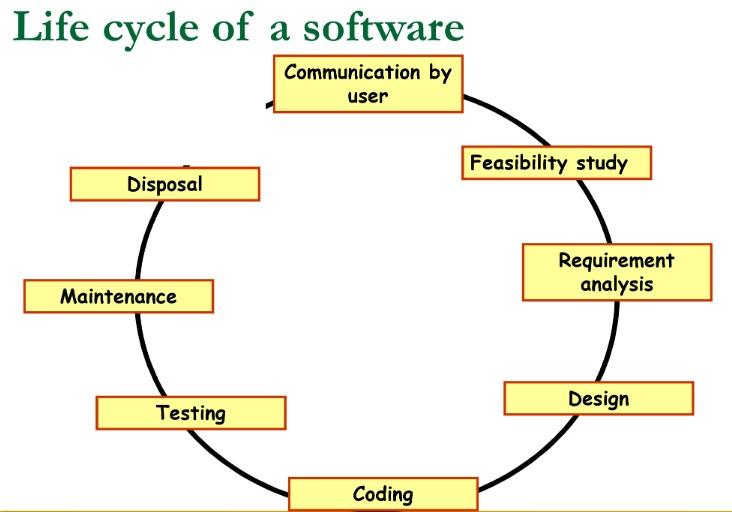
4. Dipmay Biswas 2021CSB043

5. Ketan Khandenwal 2021CSB045

*What is Software Development Life Cycle?*

The software development life cycle is a process that development teams use to create awesome software that's top-notch in terms of quality, cost-effectiveness, and time efficiency. The main goal is to minimize risks and make sure the software meets the customer's expectations both during and after production.

This process is about creating a detailed plan to guide the development of the product and then breaking down the development process into smaller modules that can be assigned, completed, and measured to make the whole thing more manageable.



A typical Software Development Life Cycle consists of the following stages −

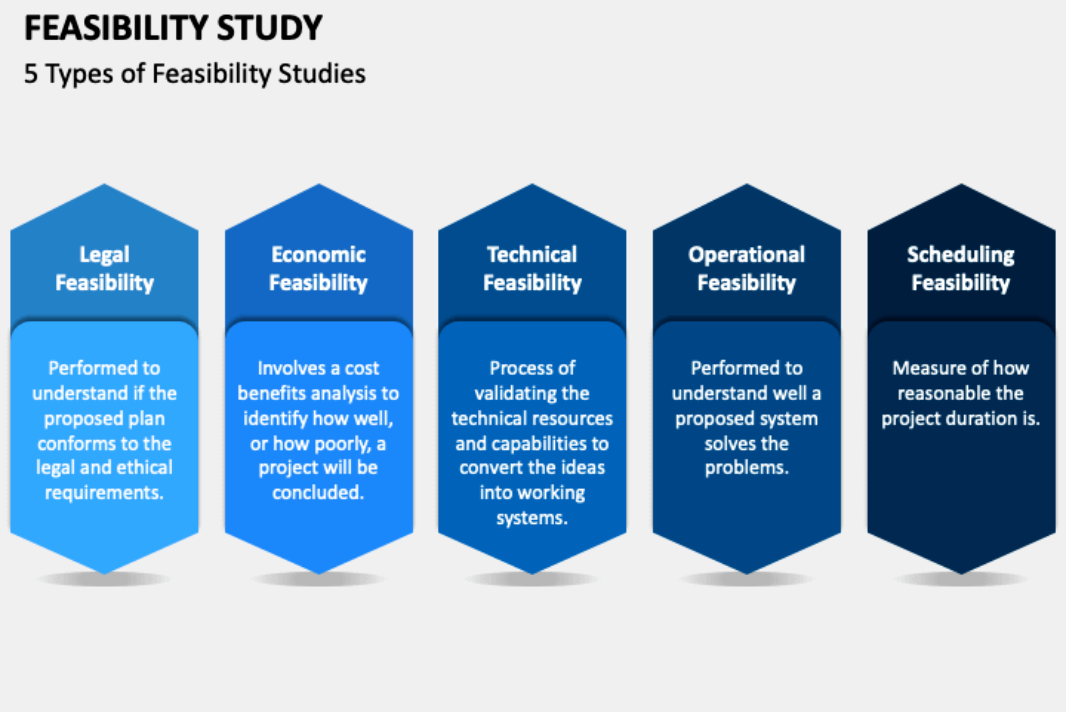
1. ***Feasibility Study:***

Feasibility study is the first step of the Software Development Life Cycle. It is a management-oriented activity that finds out if a project can be undertaken and assesses its operational, technical, and economic merit.

There are five sorts of feasibility studies - individual topics that a feasibility study investigates:

1. Technical Feasibility
2. Economic Feasibility
3. Operational Feasibility
4. Legal Feasibility
5. Scheduling Feasibility

Feasibility Study in addition to exploring all above areas generally provides some recommendations on whether a project should be undertaken or not? It also gives opinions about other alternatives. Management or client of the product, then based on the feasibility study, decide on the fate of the project.



* ***Technical Feasibility:***

A large part of determining resources has to do with assessing technical feasibility. The technical requirements of the proposed project are considered and then compared to the technical capability of the organization. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project

requirements.

It also determines if it is possible to upgrade the current system and what other

alternatives to the proposed project.

It answers the question:

● Is it a practical proposition?

● Do we currently possess the necessary technology?

● Is the project feasible within the limits of current technology?

● Manpower- programmers, testers & debuggers requirement.



***❖ Economic feasibility:***

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis, the procedure is to determine the benefits and savings expected from a candidate system and compare them with costs. If the benefits outweigh the costs, then a decision is made to design and implement the system.

Possible questions raised in economic analysis are:

● Estimated cost of hardware

● Estimated cost of software/software development·

● Is the system cost-effective?

● Do benefits outweigh costs?

● Selection among alternative financing arrangements (rent/lease/purchase)



***❖ Operational feasibility:***

Operational feasibility is a measure of how well a proposed system solves the problems

and takes advantage of the opportunities identified during scope definition and how it

satisfies the requirements identified in the requirements analysis phase of system

development. Operational feasibility reviews the willingness of the organization to

support the proposed system.

This process answers the following questions:

● Does the current mode of operation provide adequate throughput and response time?

● If the new system is developed, will it be used?

● Will the proposed system benefit the organization?

● How will the working environment of the end-user change?

● Can or will end-users and management adapt to the change?

● What end-users or managers may resist or not use the system?



***❖ Legal Feasibility:***

This evaluation looks at whether any component of the planned project violates any regulations,

such as zoning regulations, data protection legislation, or social media legislation.



**❖ Scheduling Feasibility:**

An organization predicts the length of time it will take to finish a project in scheduling

feasibility. Generally, projects are also divided into different phases, and deadlines for

each phase are provided separately so that resources can be mapped to them efficiently.

***CASE STUDY ON FEASIBILITY STUDY:***

#### **Starting a New Family Restaurant:**

Conducting a feasibility study before opening the restaurant will help the owner save time and money as, with the help of a study, he can make an informed decision regarding the chance of success of the venture.

In the present case, the following are the different factors that the feasibility study will include:

* **Obtain Market Statistics:** Feasibility study should include studying the demographic characteristics like age and income to know the size of the potential market. In the case of the family restaurant, one should know the number of families residing in the area as singles or students will not count for the potential share.
* **Potential Locations:** The location for the family restaurant should be an area having traffic. Parking and other factors should also be considered to make sure that the place is easily accessible to customers. Also, there should be a proper tradeoff between location and [lease](https://www.wallstreetmojo.com/lease/) cost.
* **Competition:** At the time of the Feasibility study, one should gather information about the total number of nearby restaurants and the style of those restaurants. The area should not already be saturated with a similar concept as planned. Thus one should properly analyze the strengths and weaknesses of all major competitors.
* **Industry Analysis:** To study the feasibility, one should join hospitality organizations and attend their meetings to learn more about the health and growth of the industry.
* **Current Economic Environment:** Decide whether as per the current economic environment launching a new restaurant is advisable or not. Whether any restaurants were closed in the past few years or not and their reasons thereof.
* **Cost Structure:** One should break down the cost of each item on the menu and determine the major suppliers in the future and the prices offered by them. Also, there should be a proper cost projection of the food cost projections.

In the feasibility study, one should evaluate management capability, i.e., whether an entrepreneur has the required skill and experience to make a venture successful. He should also have the capability to manage the staff and to understand properly about kitchen operation.

***2. Requirement Analysis:***

Requirements analysis or requirements engineering is a process used to determine the needs and expectations of a new product. It involves frequent communication with the [stakeholders](https://www.simplilearn.com/stakeholders-impact-on-the-projects-article) and end-users of the product to define expectations, resolve conflicts, and document all the key requirements.

A requirements analysis process involves the following steps:

### ***Step 1: Identify Key Stakeholders and End-Users***

### ***Step 2: Capture Requirements:***

1. Hold One-On-One Interviews

2. Use Focus Groups

3. Utilize Use Cases

4. Build Prototypes

### ***Step 3: Categorize Requirements:***

1. Functional Requirements: Functions the product is required to perform.

2. Technical Requirements: Technical issues to be considered for the successful implementation of the product.

3. Transitional Requirements: Steps required to implement a new product smoothly.

4. Operational Requirements: Operations to be carried out in the backend for proper functioning of the product.

### 

### ***Step 4: Interpret and Record Requirements:***

1. Define Requirements Precisely

#### 2. Prioritize Requirements

#### 3. Carry Out an Impact Analysis

#### 4. Resolve Conflicts

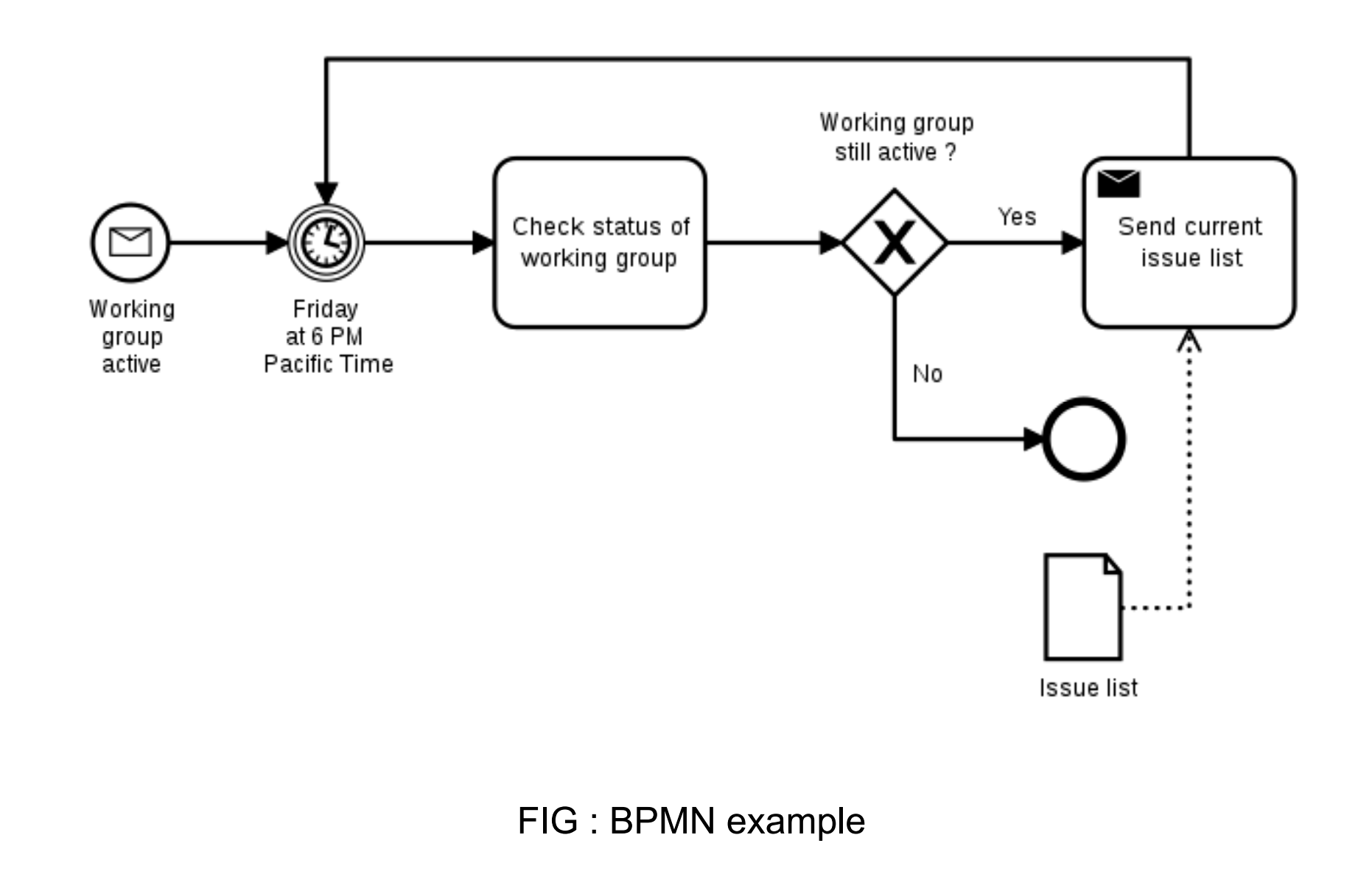
#### 5. Analyze Feasibility

### ***Step 5: Sign off***

## ***Requirement Analysis Techniques:***

### ***1. Business Process Model and Notation (BPMN):***

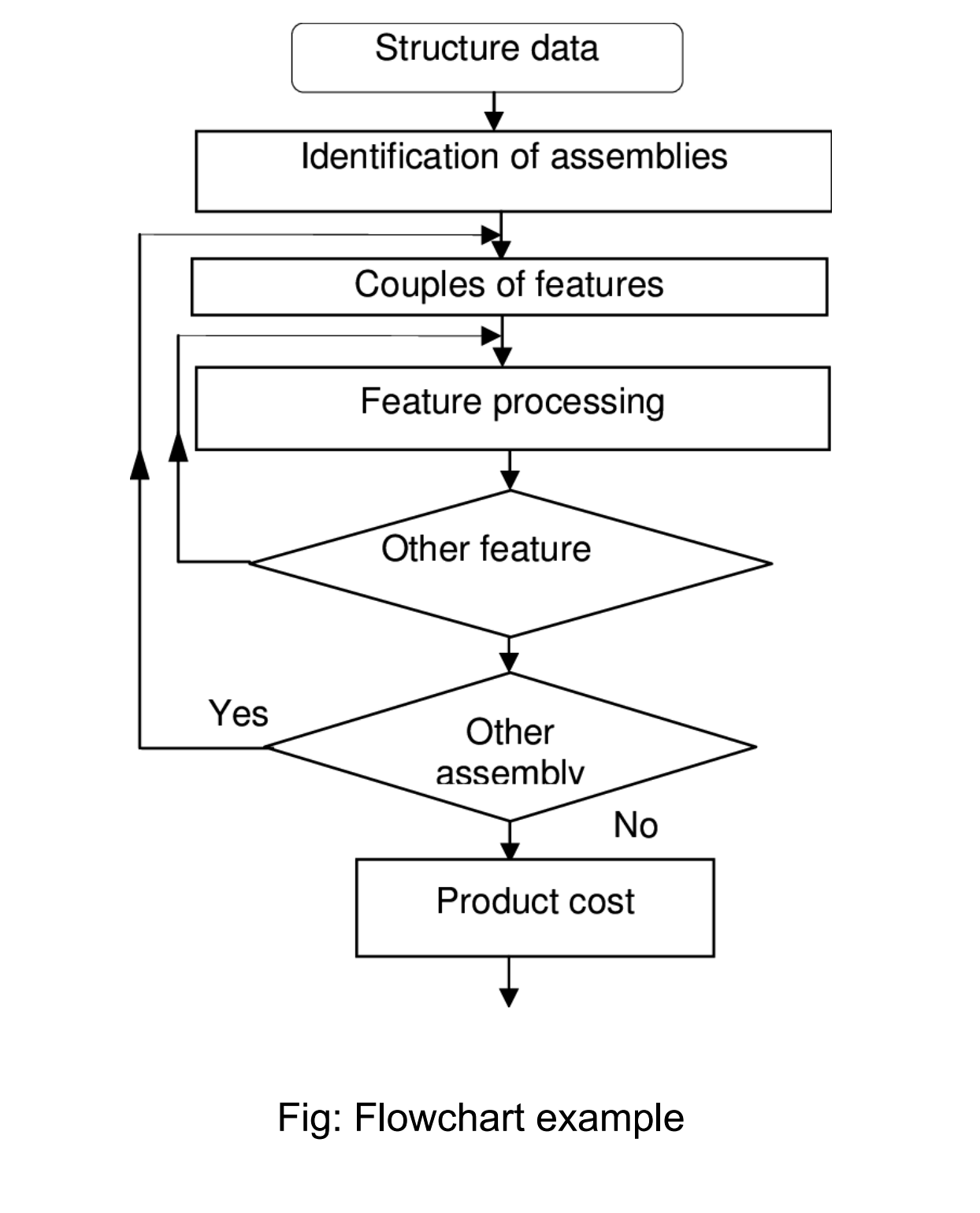
Business Process Model and Notation is used to create graphs that simplify the understanding of the business process. It is a popular technique used by [business analysts](https://www.simplilearn.com/what-is-business-impact-analysis-article) to coordinate the sequence of messages between different participants in a related set of activities.



### 

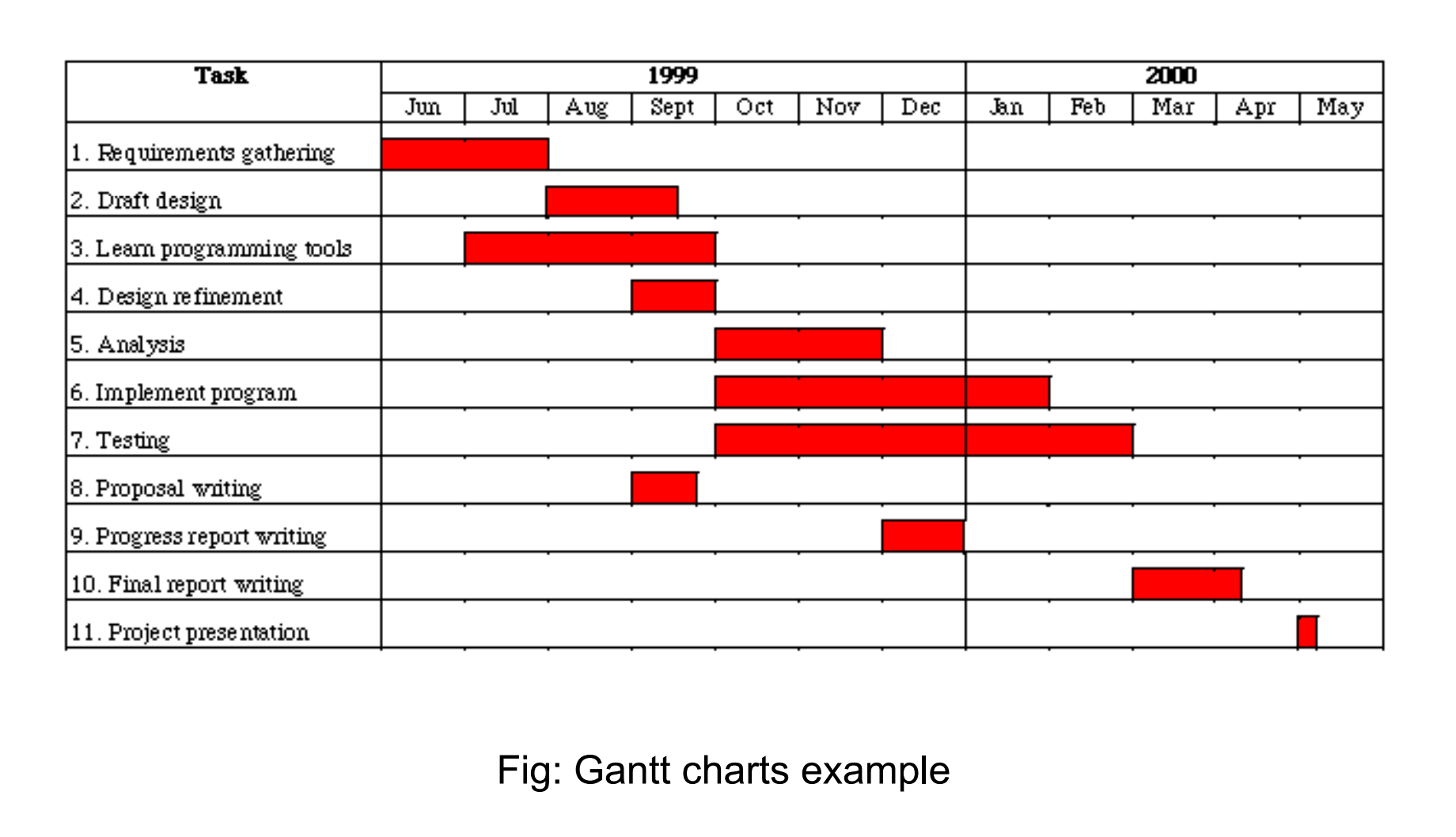
### ***2. Flowcharts:***

Flowcharts depict sequential flow and control logic of a related set of activities. They are useful for both technical and non-technical members.



### ***3. Gantt Charts:***

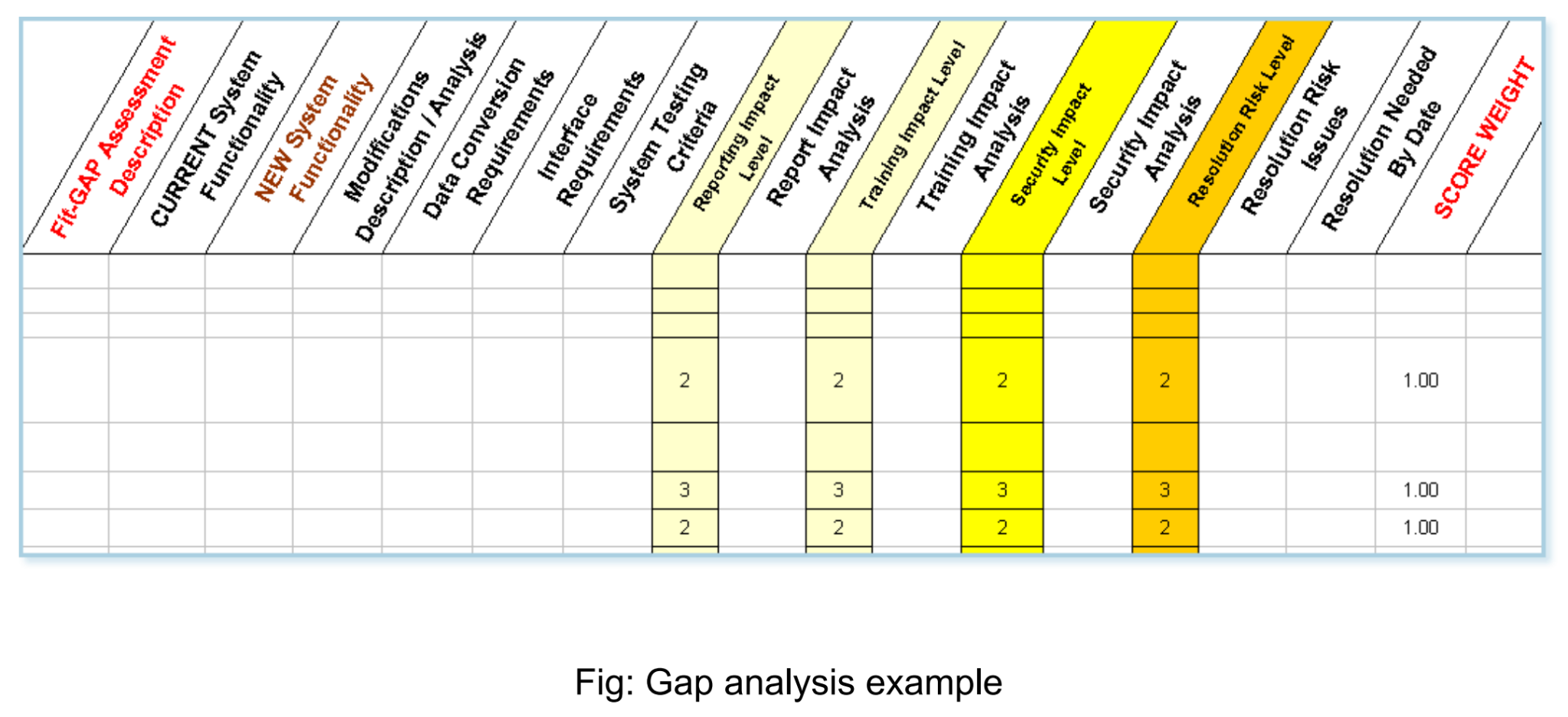
[Gantt Charts](https://www.simplilearn.com/what-is-a-gantt-chart-article) provide a visual representation of tasks along with their scheduled timelines. They help business analysts visualize the start and end dates of all the tasks in a project.



### 

### ***4. Gap Analysis***

evaluates the gaps in a product’s performance to determine whether the requirements are met or not. They help [business analysts](https://www.simplilearn.com/business-analyst-career-path-article) determine the present state and target state of a product.



***CASE STUDY ON REQUIREMENT ANALYSIS:***

**Project Name:** ***A Fictional Online E-Commerce Website***

***Objective: Create a user-friendly e-commerce website to allow customers to browse and purchase products online.***

***Stakeholders:***

1. Customers (shoppers)
2. Product vendors and sellers
3. Website administrators
4. Payment gateway providers
5. Shipping and logistics partners

***unctional Requirements:***

***1. User Registration and Authentication:***

* Customers must be able to create accounts and log in.
* User authentication should be secure.
* Provide an option for social media login.

***2. Product Catalog:***

* Display a catalog of products, categorized by type and brand.
* Include high-quality images, detailed descriptions, and prices for each product.
* Allow customers to search for products by keyword and filter results.

***3. Shopping Cart:***

* Enable customers to add and remove items from their shopping cart.
* Display the total cost of items in the cart.
* Save cart contents between sessions for registered users.

***4. Checkout and Payment:***

* Provide a secure checkout process with multiple payment options (credit/debit cards, PayPal, etc.).
* Collect shipping and billing information from customers during checkout.
* Apply discounts, coupons, and gift cards if available.
* Send order confirmation emails to customers.

***5. Order History:***

* Store order history for registered users.
* Allow users to view and track the status of their orders.

***6. Product Reviews and Ratings:***

* Allow registered customers to leave reviews and ratings for products.
* Display average ratings and user-generated reviews on product pages.

***7. User Profiles:***

* Customers can edit their profiles, including shipping addresses and contact information.
* Store multiple shipping addresses for registered users.

***8. Inventory Management:***

* Update product availability in real-time.
* Notify administrators when products are out of stock.

***9. Admin Panel:***

* Provide an admin panel for managing products, orders, and user accounts.
* Support role-based access control for administrators.

***Non-Functional Requirements:***

***1. Performance:***

* Ensure fast loading times, even with a high number of concurrent users.
* Optimize website performance for mobile devices.

***2. Security:***

* Implement secure encryption for user data and payment information.
* Protect against common web application vulnerabilities (e.g., SQL injection, cross-site scripting).

***3. Usability:***

* The website should have an intuitive and responsive design.
* Support multiple languages and accessibility features.

***4. Scalability:***

* Design the website to handle increased traffic during peak periods (e.g., holiday sales).

***5. Data Backup and Recovery:***

* Regularly backup customer data and order history.
* Implement a disaster recovery plan.

**Constraints:**

***1. Budget:***

* The project has a predefined budget for development and maintenance.

***2. Legal Compliance:***

* Ensure compliance with data protection regulations (e.g., GDPR, CCPA).
* Adhere to e-commerce regulations and consumer protection laws.

**Assumptions:**

1. Payment gateway providers will integrate seamlessly with the website for processing payments.

2. Product vendors will provide accurate and up-to-date product information.

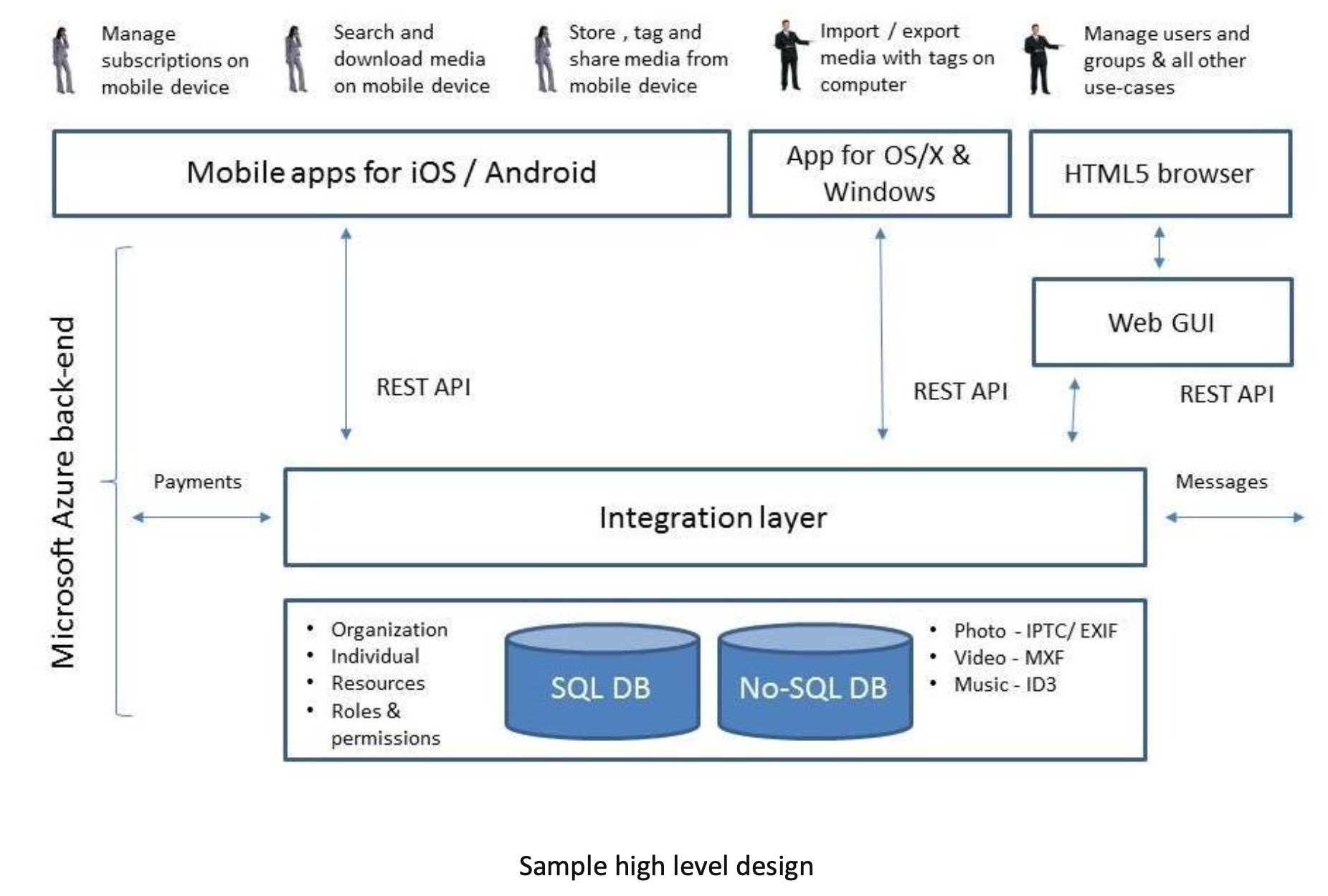
***3. DESIGN :***

## **What Is The Design Phase in SDLC?**

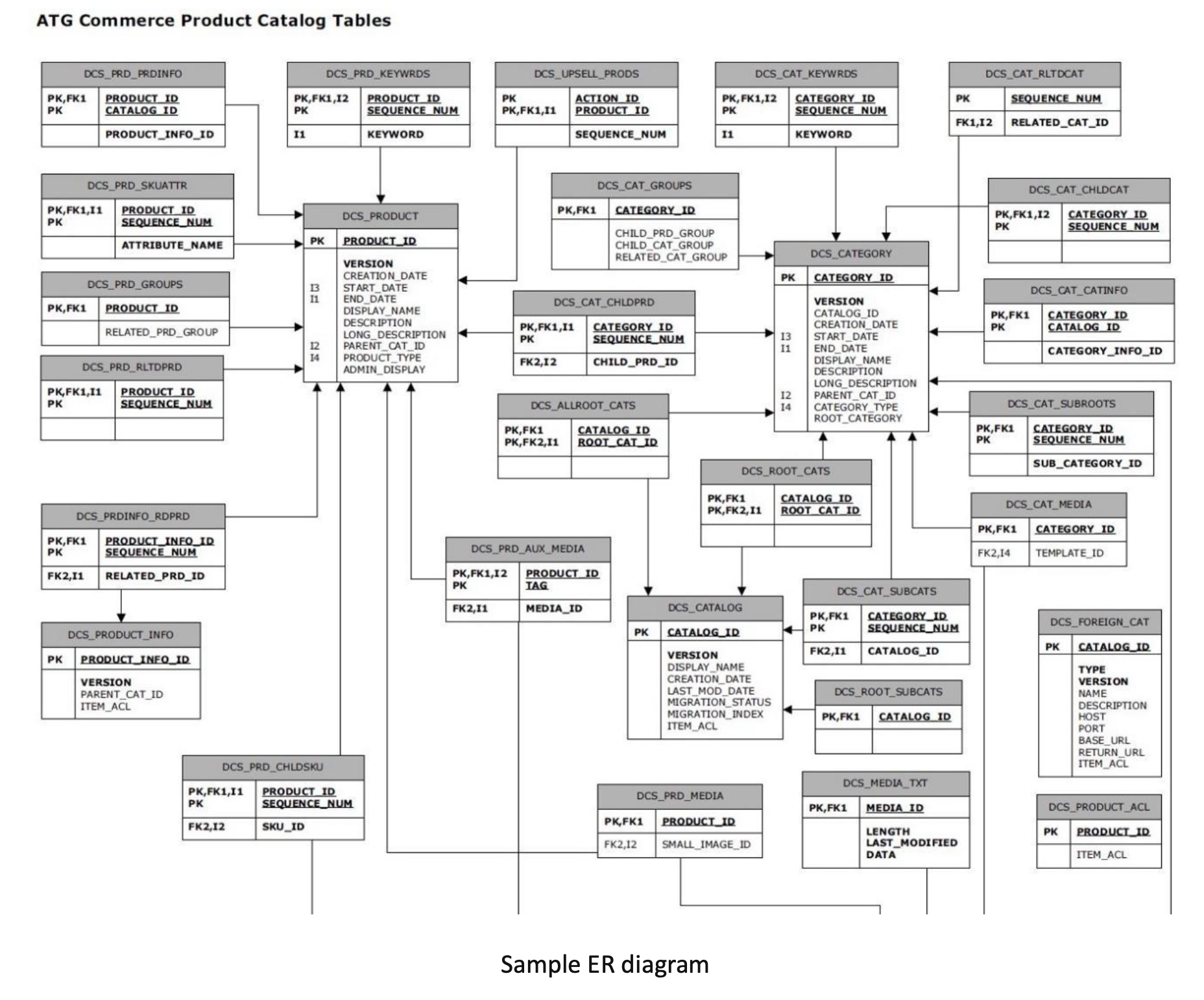
In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture. This design phase serves as input for the next phase of the model.

There are two kinds of designs involved –

1. High level design – HLD contains a brief description of modules. It just gives an outline of the functionalities of each module and the relationship between different modules. It describes the overall architecture of the system.



2. Low-level design – LLD involves defining the functional logic of the modules. In this stage, the database designs are created including field types, names, and sizes. Usually, an ER diagram is created. It describes a highly detailed structure of the software and all the details, relationships, flows, API details, dependencies, error messages, and interfaces involved.



***4. Coding:***

Coding is the process of transforming the design of a system into a computer language format. This software development coding phase concerns software translating design specifications into the source code. It is necessary to write source code & internal documentation so that conformance of the code to its specification can be easily verified.

Coding is done by the coder or programmers who are independent people rather than the designer. The goal is not to reduce the effort and cost of the coding phase but to cut to the cost of a later stage. The cost of testing and maintenance can be significantly reduced with efficient coding.

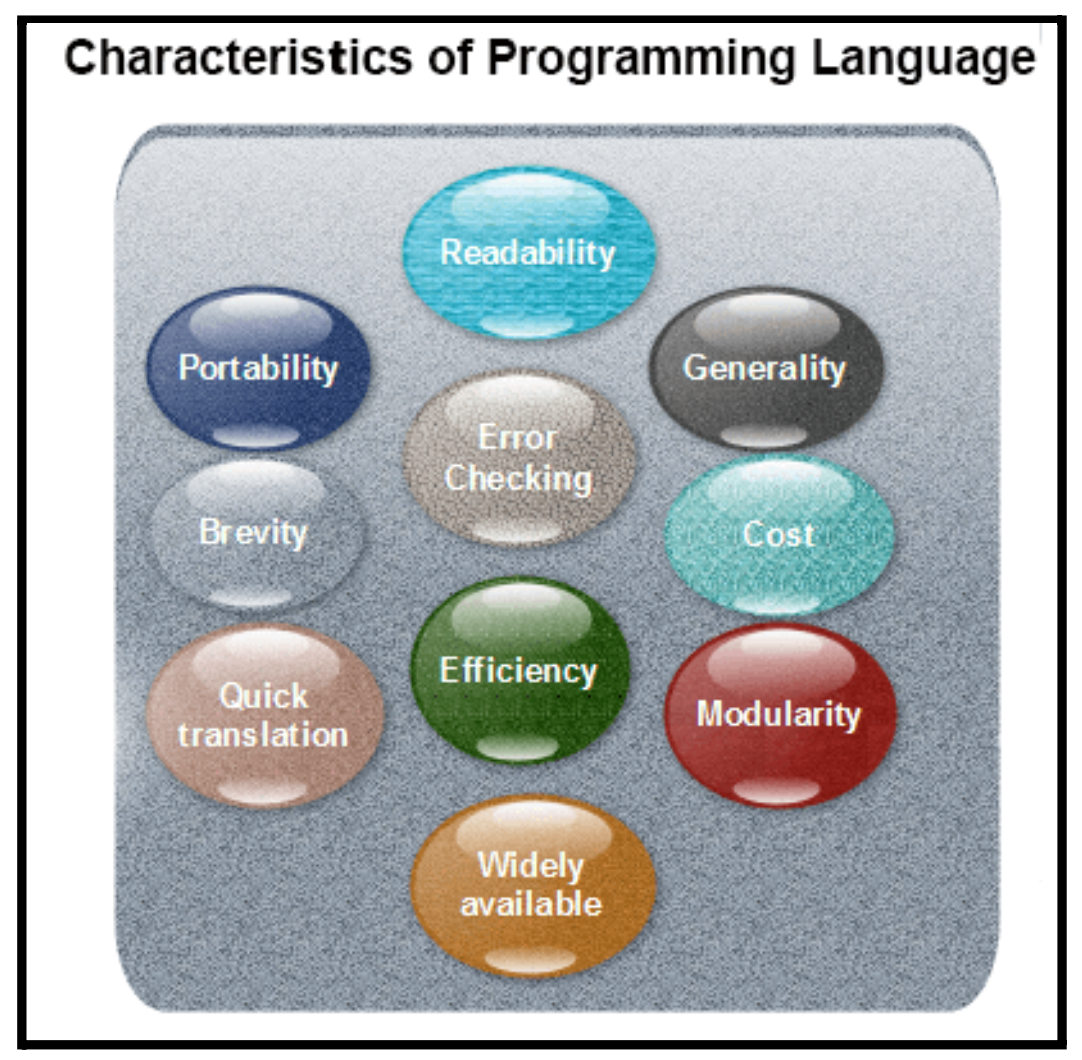
***Goals of Coding***

➢ To translate the design of the system into a computer language format

➢ To reduce the cost of later phases  
➢ Making the program more readable

*To implement our design into code, we require a high-level functional language.*

***❖ Characteristics of Programming Language :***



***Readability:*** A good high-level language will allow programs to be written in some methods that resemble a quite English description of the underlying functions. The coding may be done in an essentially self-documenting way.

***Portability:*** High-level languages, being virtually machine-independent, should be easy to develop portable software.

***Generality:*** Most high-level languages allow the writing of a vast collection of programs, thus relieving the programmer of the need to develop into an expert in many diverse languages.

***Brevity:*** Language should have the ability to implement the algorithm with less amount of code. Programs in high-level languages are often significantly shorter than their low-level equivalents.

***Error checking:*** A programmer is likely to make many errors in the development of a computer program. Many high-level languages invoke a lot of bugs checking both at compile-time and run-time.

***Cost:*** The ultimate cost of a programming language is a task of many of its characteristics.

***Quick translation:*** It should permit quick translation.  
***Efficiency:*** It should authorize the creation of an efficient object code.

***Modularity:*** It is desirable that programs can be developed in the language as several separately compiled modules, with the appropriate structure for ensuring self-consistency among these modules.

***Widely available:*** Language should be widely available, and it should be feasible to provide translators for all the major machines and all the primary operating systems.

***❖ Coding Standards:***

A coding standard lists several rules to be followed during coding, such as the way variables are to be named, the way the code is to be laid out, error return conventions, etc.

General coding standards refer to how the developer writes code, so here we will discuss some essential standards regardless of the programming language being used.

***1. Indentation:*** Proper and consistent indentation is essential in producing easy-to-read and maintainable programs.  
 Indentation should be used to:

o Emphasize the body of a control structure such as a loop or a select statement. o Emphasize the body of a conditional statement  
 o Emphasize a new scope block

***2. Inline comments:*** Inline comments analyze the functioning of the subroutine, or key aspects of the algorithm shall be frequently used.

***3. Rules for limiting the use of global:*** These rules file what types of data can be declared global and what cannot.

***4. Structured Programming:*** Structured (or Modular) Programming methods shall be used. "GOTO" statements shall not be used as they lead to "spaghetti" code, which is hard to read and maintain, except as outlined in the FORTRAN Standards and Guidelines.

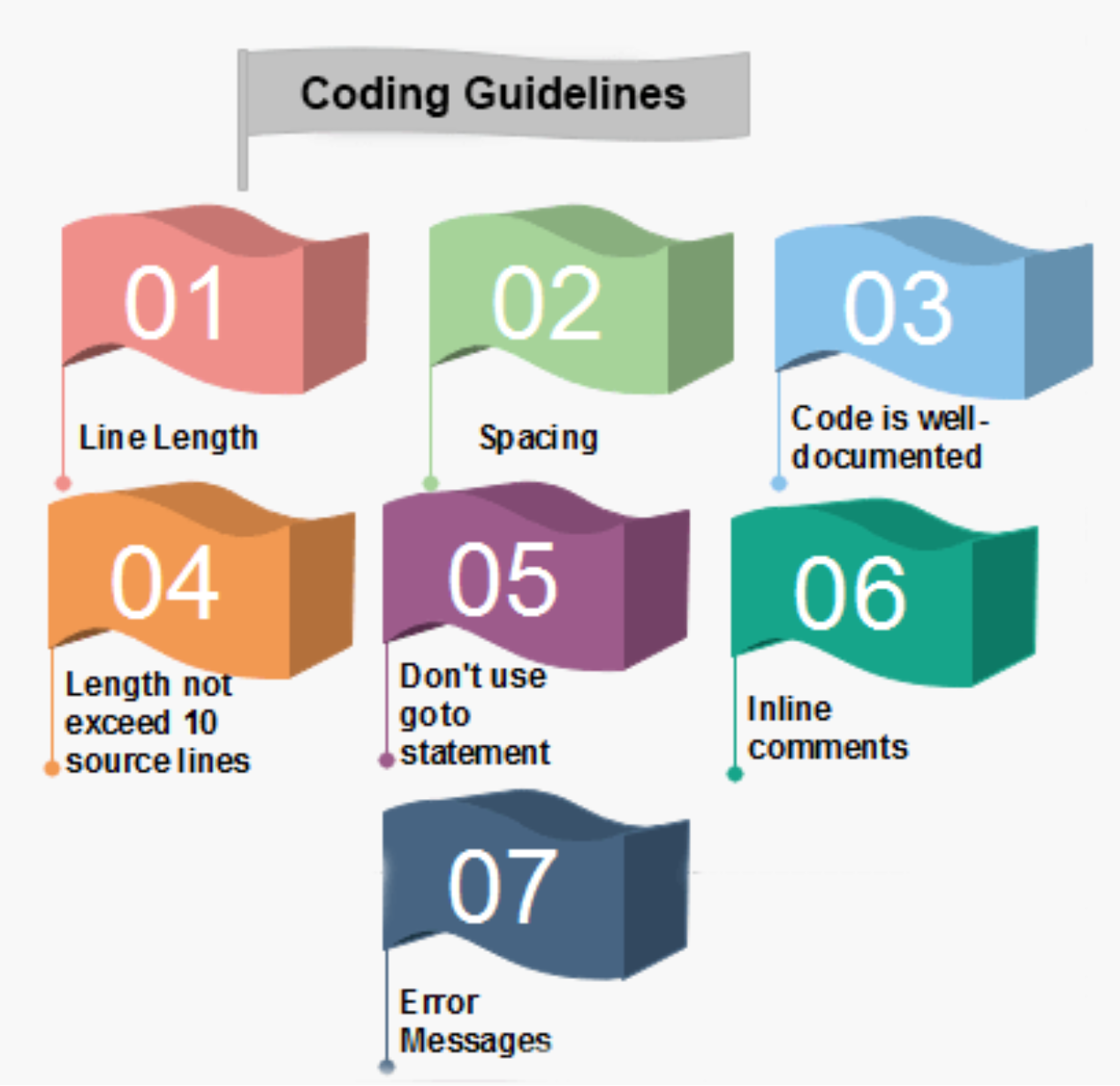
***5. Naming conventions for global variables, local variables, and constant identifiers:*** A possible naming convention can be that global variable names always begin with a capital letter, local variable names are made of small letters, and constant names are always capital letters.

***6. Error return conventions and exception handling system:*** Different functions in a program report the way error conditions are handled should be standard within an organization. For example, different tasks while encountering an error condition should either return a 0 or 1 consistently.

***❖ Coding Guidelines:***

General coding guidelines provide the programmer with a set of the best methods that can be used to make programs more comfortable to read and maintain. Most of the examples use the C language syntax, but the guidelines can be tested in all languages.

The following are some representative coding guidelines recommended by many software development organizations.



***1. Line Length:*** It is considered good practice to keep the length of source code lines at or below 80 characters. Lines longer than this may not be visible properly on some terminals and tools. Some printers will truncate lines longer than 80 columns.

***2. Spacing:*** The appropriate use of spaces within a line of code can improve readability.

***Example:***

**Bad:** cost=price+(price\*sales\_tax)  
 fprintf(stdout ,"The total cost is %5.2f\n",cost);

**Better:** cost = price + ( price \* sales\_tax )  
 printf (stdout,"The total cost is %5.2f\n",cost);

***3. The code should be well-documented:*** As a rule of thumb, there must be at least one comment line on the average for every three-source line.

***4. The length of any function should not exceed 10 source lines:*** A very lengthy function is generally very difficult to understand as it possibly carries out many various functions. For the same reason, lengthy functions are possible to have a disproportionately larger number of bugs.

***5. Do not use goto statements:*** The use of goto statements makes a program unstructured and very tough to understand.

***6. Inline Comments:*** Inline comments promote readability.

***7. Error Messages:*** Error handling is an essential aspect of computer programming. This does not only include adding the necessary logic to test for and handle errors but also involves making error messages meaningful.

***EXAMPLE:***

***Sample code for designing a login page:-***



***5. Testing:***

The testing phase of the software development lifecycle (SDLC) is where you focus on investigation and discovery. During the **testing phase**, developers find out whether their code and programming work according to customer requirements. While it's not possible to solve all the failures you might find during the testing phase, it is possible to use the results from this phase to reduce the number of errors within the software program.

Before testing can begin, the project team develops a **test plan**. The test plan includes the types of testing you'll be using, resources for testing, how the software will be tested, who should be the testers during each phase, and **test scripts**, which are instructions each tester uses to test the software. Test scripts ensure consistency while testing.

There are several types of testing during the test phase, including quality assurance testing (QA), system integration testing (SIT), and user acceptance testing (UAT).

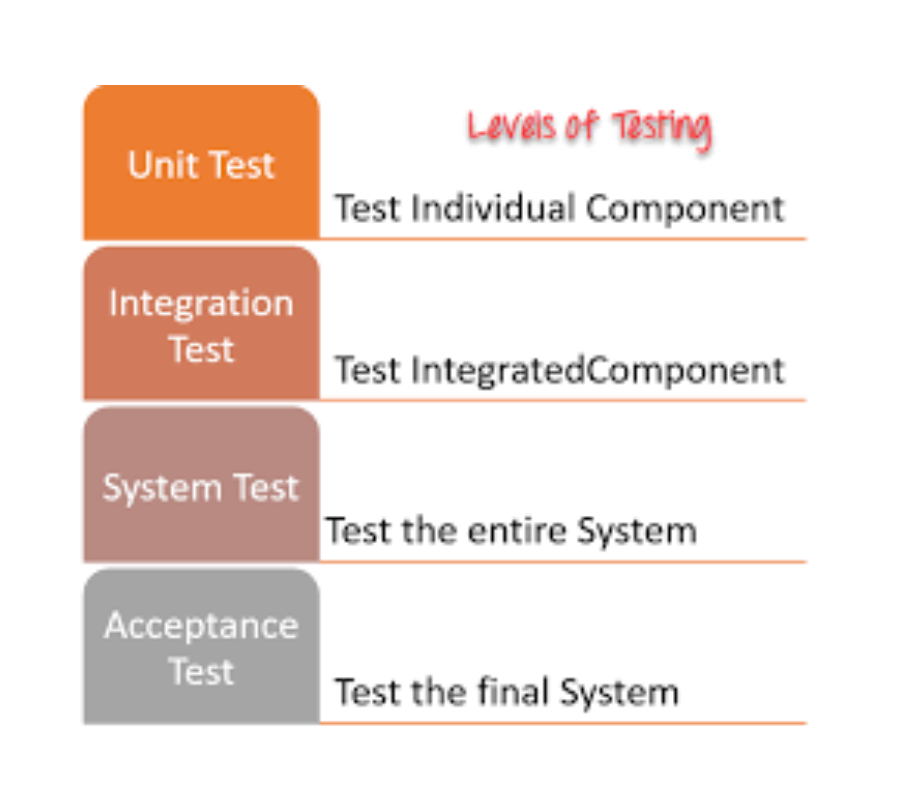
Four major types of tests are –

***1. Unit testing –*** During this first round of testing, the program is submitted to assessments that focus on specific units or components of the software to determine whether each one is fully functional. The main aim of this endeavor is to determine whether the application functions as performing as expected.

***2. Integration testing –*** Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to **find interface defects between the modules/functions**.

***3. System testing –*** System testing is the first level in which **the complete application is tested as a whole**. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards.

***4. Acceptance testing –*** The final level, Acceptance testing (or User Acceptance Testing), is conducted to **determine whether the system is ready for release**.



***6. Maintenance:***

After all the above processes of Software Development, the Product is in public, and people are using it. There will be situations where users might face problems in terms of bad User experience or some Program Bugs. Fixing those issues is necessary to satisfy user expectations and experience. This is achieved by performing maintenance. This is the phase for all the future revisions of the product to fix bugs, improve software design, enhance the product, accommodate new features and hardware, etc.

***❖ Categories of Software Maintenance:***

***1. Corrective Maintenance:***

* + Rectify/Correct bugs in the product found by users
  + Enhance the performance of the product

***2. Adaptive Maintenance:***

a. Updates to the software to allow products to run in new environments.

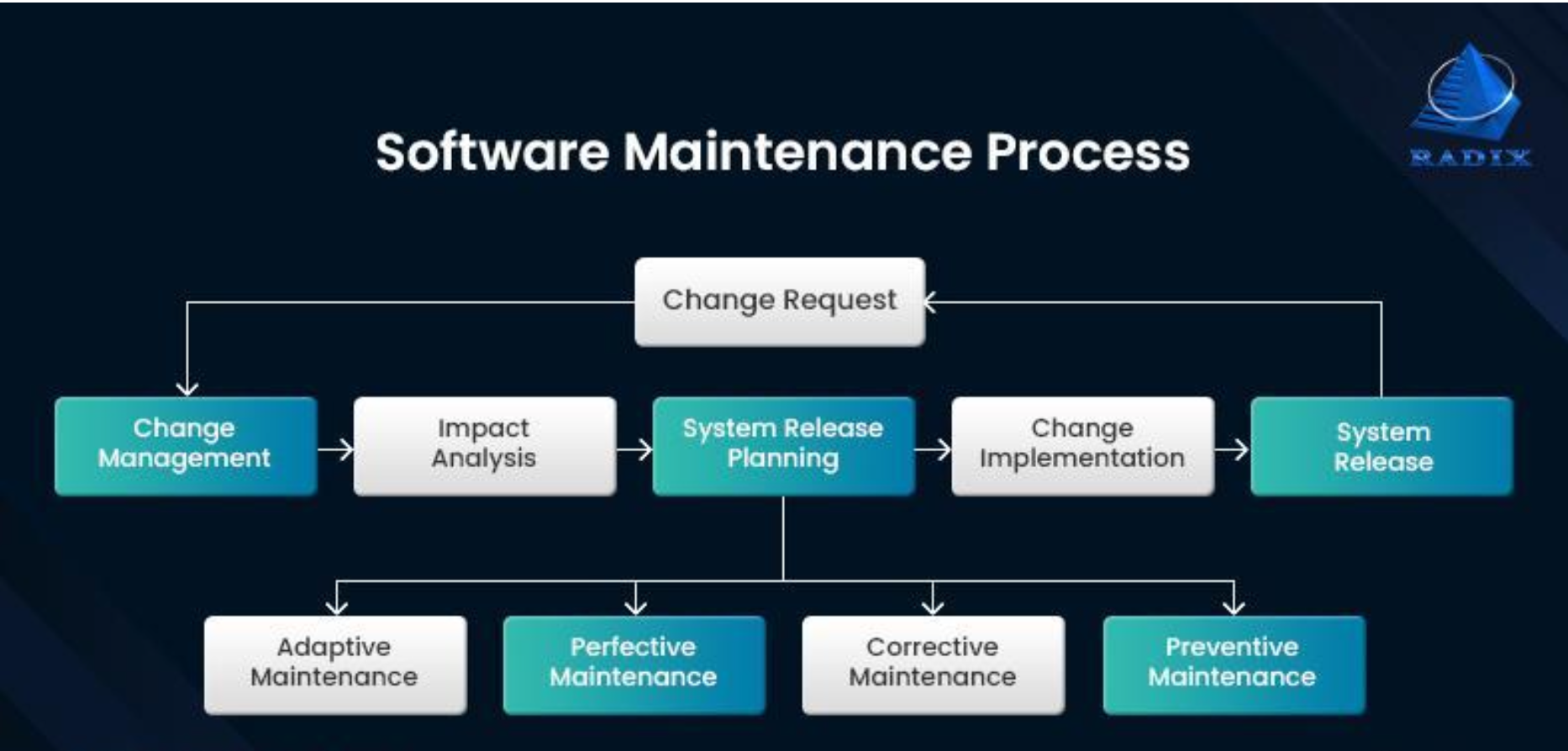
***3. Perfective Maintenance***

a. New Features for the product for the users

***4. Preventive Maintenance***

a. Fixes to prevent problems in the product in the future.

***❖ Case Study:***

* What this is:  
   i. A Beta testing program on which Microsoft, as a company, can evaluate their future versions of the Windows Operating System.
* How it works  
  1. Users who want to try beta testing in Windows can sign up for this program to get unstable versions of the Operating System and can take part in the testing and maintenance of Windows.
  2. If any bugs and inconsistencies are found in the operating system, the users can send their reports in an app named “Feedback Hub” which sends the reports directly to the Microsoft Windows Development Division
  3. From there, if the reports are valid, engineers at Microsoft fix those issues.
  4. They also provide new releases for users to evaluate and generate any error reports.
  5. This entire process of bug reports and fixing happens weekly and for months, providing a legitimate and stable version of Windows that can be pushed as an update to the public. 

***REFERENCES:***

*REFERENCES FOR FEASIBILITY STUDY:*

1. Feasibility Study in SDLC - FreeEducator.com
2. Feasibility study
3. Sample Software Engineering Feasibility Study Report

*REFERENCES FOR REQUIREMENT ANALYSIS:*

1. SRSExample-webapp.doc (live.com)
2. SRS4.0.doc (live.com)
3. Requirements Document Example
4. BalticLSC Software User Requirements Specification

*REFERENCES FOR DESIGN:*

1. Example XML Legal Document Utility Software Design Document

*REFERENCES FOR CODING:*

1. C++ Program to Create Login Page - Quiz for Exam
2. Software Engineering | Coding - javapoint

*REFERENCES FOR TESTING:*

1. Manual Testing Tutorial: What is, Concepts, Types & Tool
2. Unit Testing Tutorial: What is, Types, Tools & Test EXAMPLE