SRI AKILANDESWARI WOMEN'S COLLEGE, WANDIWASH

SOFTWARE ENGINEERING

CLASS: III B.Com (CA)

S.APPANDAI RAJAN,

ASSISTANT PROFESSOR,

DEPARTMENT OF COMPUTER APPLICATIONS.

SWAMY ABEDHANDHA EDUCATIONAL TRUST, WANDIWASH

LECTURE CONTENTS

SECTION-1: Basic Concepts

SECTION-2: Software Process Models

SECTION-3: Requirement Engineering Process

SECTION-4: Data Engineering

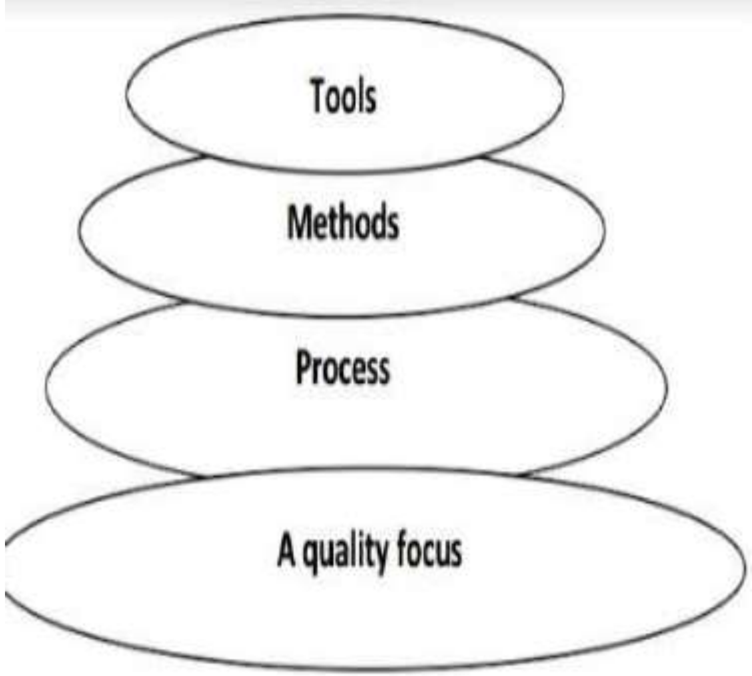
SECTION-5: Testing & Project Management

DEFINITION

- Software Engineering is the discipline of designing, building and maintaining software systems using systematic approaches, methodologies and tools to ensure the reliability, efficiency and maintainability of the software.
- Software Engineering is the establishment and use of sound engineering principles to obtain economically software that is reliable and works efficiently on real machines.

Software Engineering -A layered Technology

- Application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software that is, the application of engineering software.



Software Engineering Layers

Generic Process Framework Activities

The following generic process framework is applicable to the majority of software projects.

- Communication.
- Planning.
- Modeling.
- Construction.
- Deployment.

Software Process Models

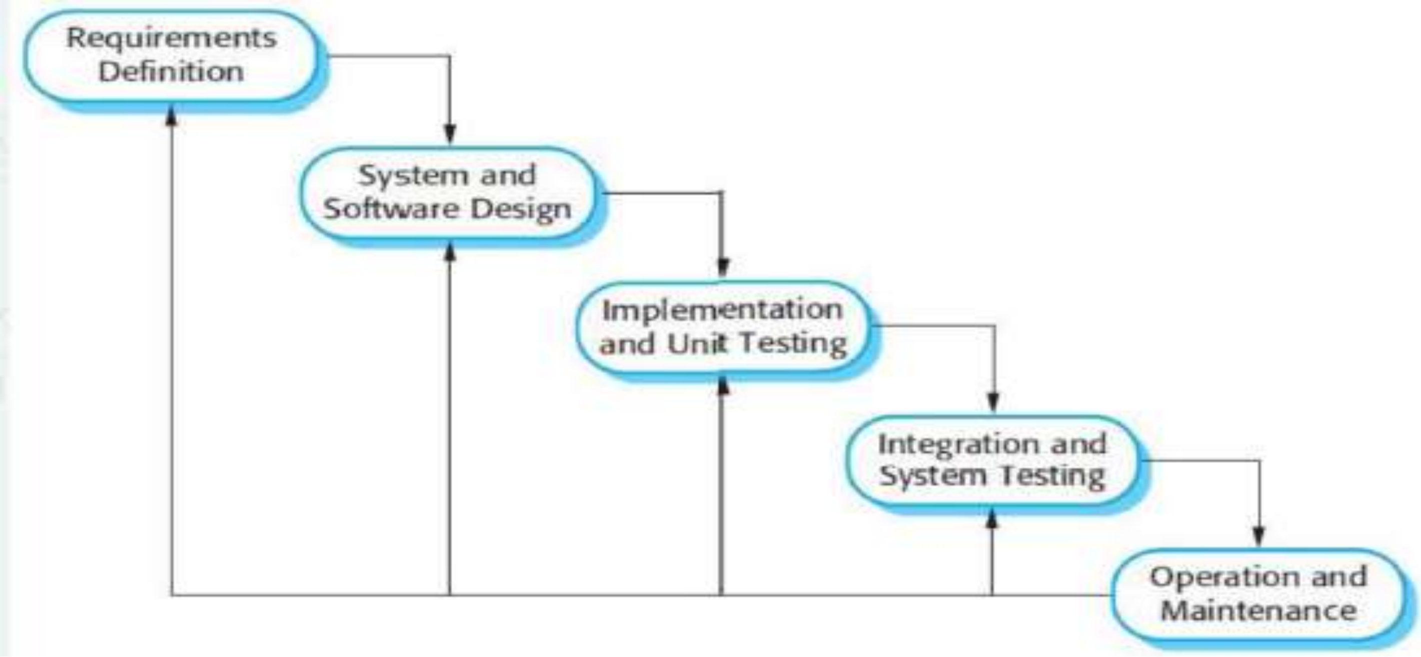
A software process model is a framework that outlines the tasks, activities, and deliverables needed to develop high-quality software. It serves as a roadmap for software development teams, providing a step-by-step guide.

Different types of Software Models are given below:

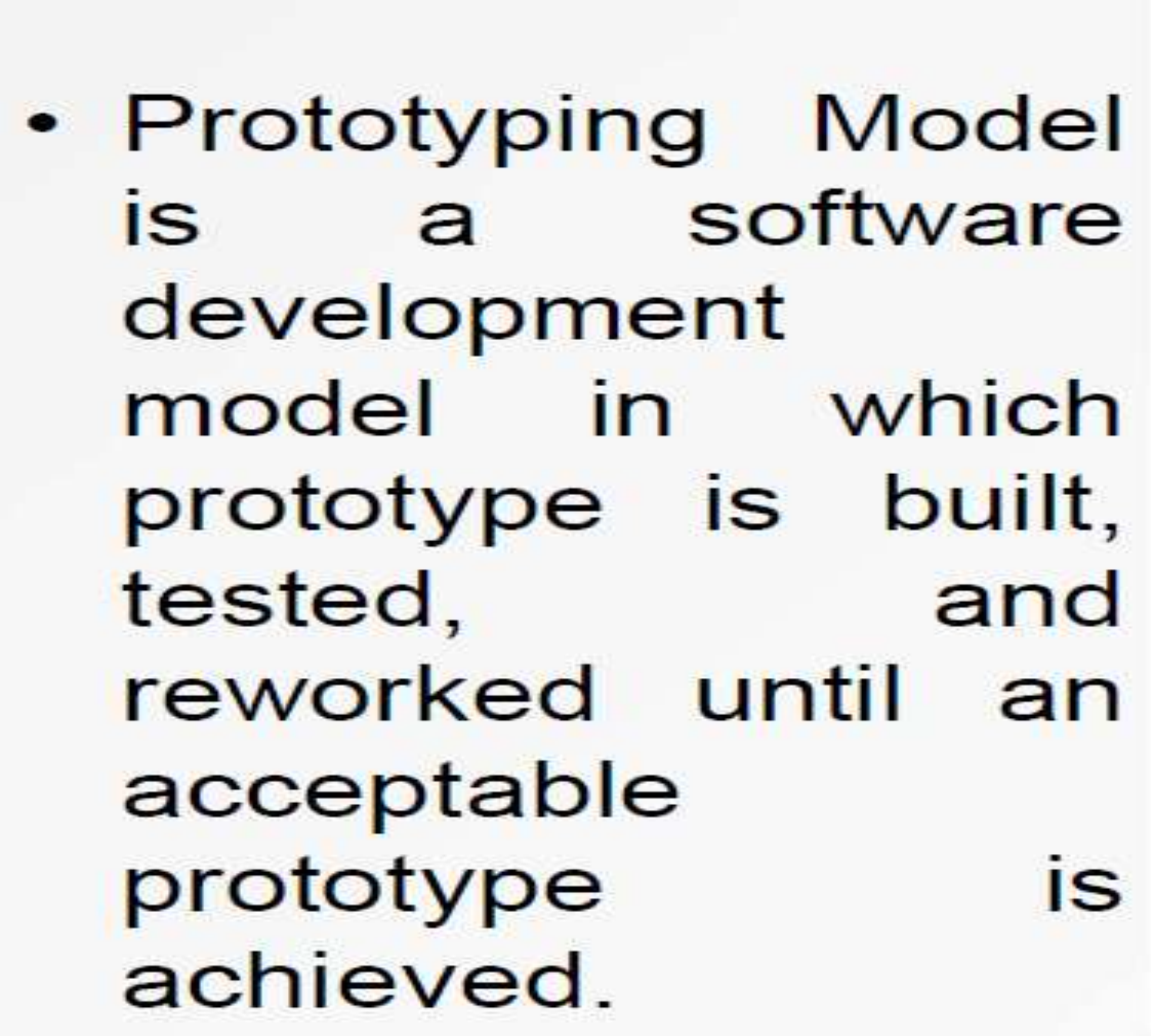
- Waterfall Life Cycle Model.
- Iterative Waterfall Life Cycle Model.
- Prototyping Model.
- Incremental Model.
- Sprial Model.
- RAD Model.

Waterfall Life Cycle Model.

- It is also called as classic life cycle or Linear model.
- Requirements are well defined and stable and It suggests a systematic, sequential approach to software development.
- It begins with customer specification of requirements and progresses.
- Stages are Planning, Modeling, Construction and Deployment.


Requirements Definition
System and Software Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

PROTOTYPE MODEL



- Prototyping Model is a software development model in which prototype is built, tested, and reworked until an acceptable prototype is achieved.

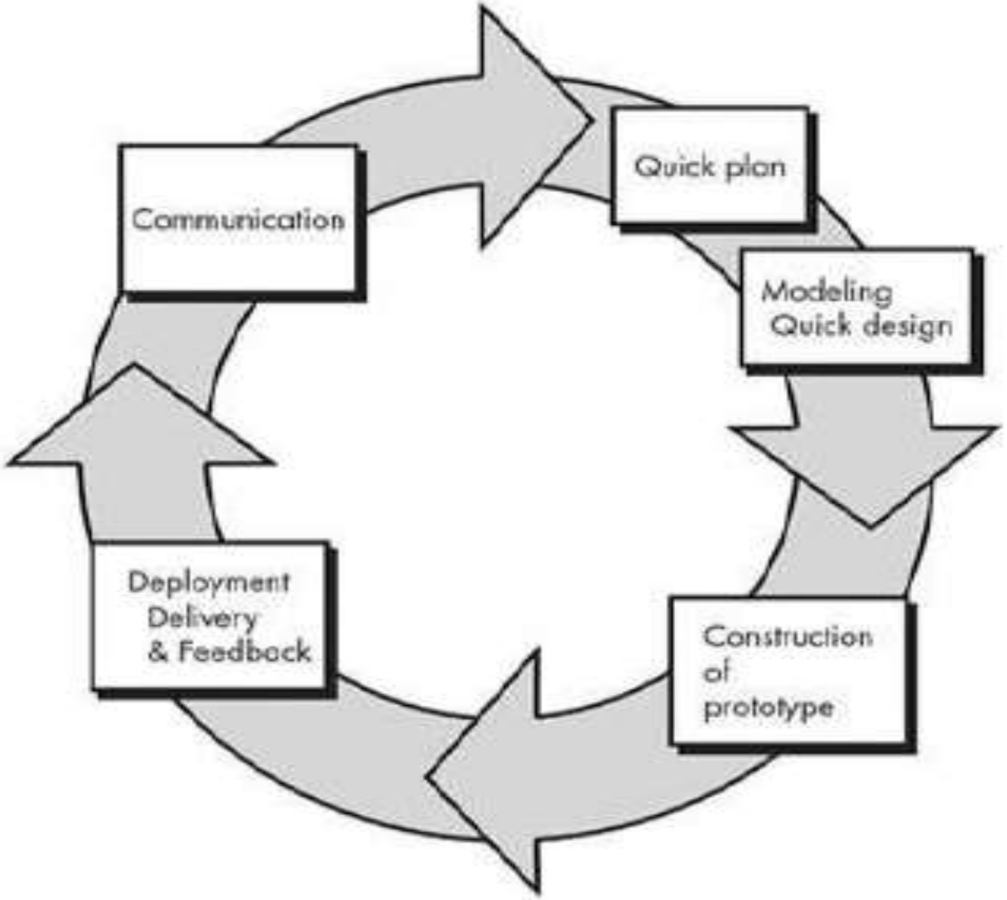
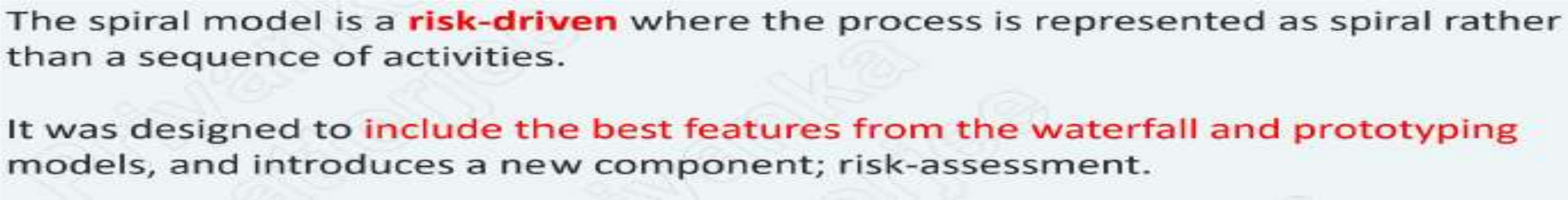


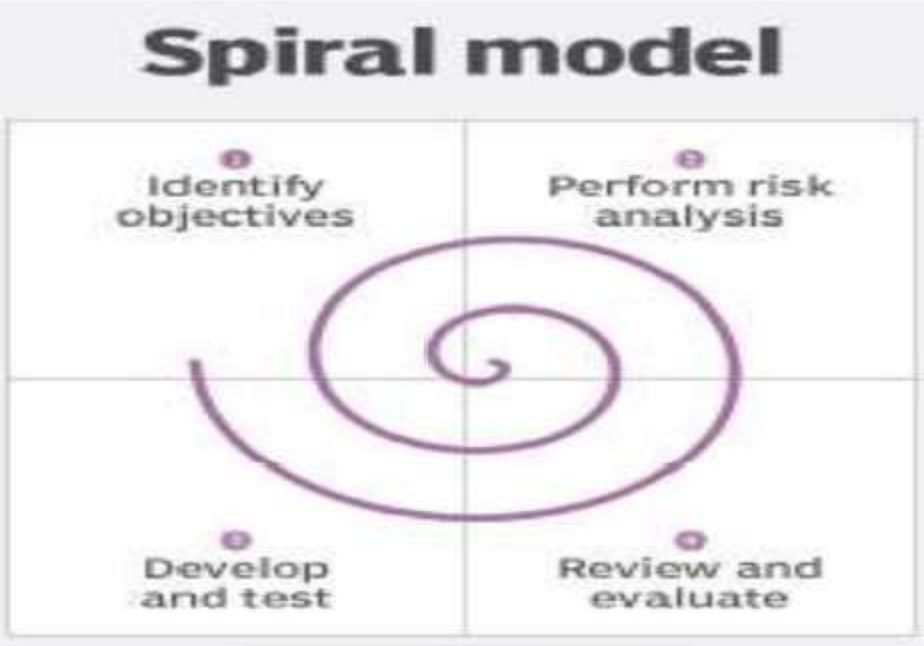
Figure: Prototype Model

SPIRAL MODEL



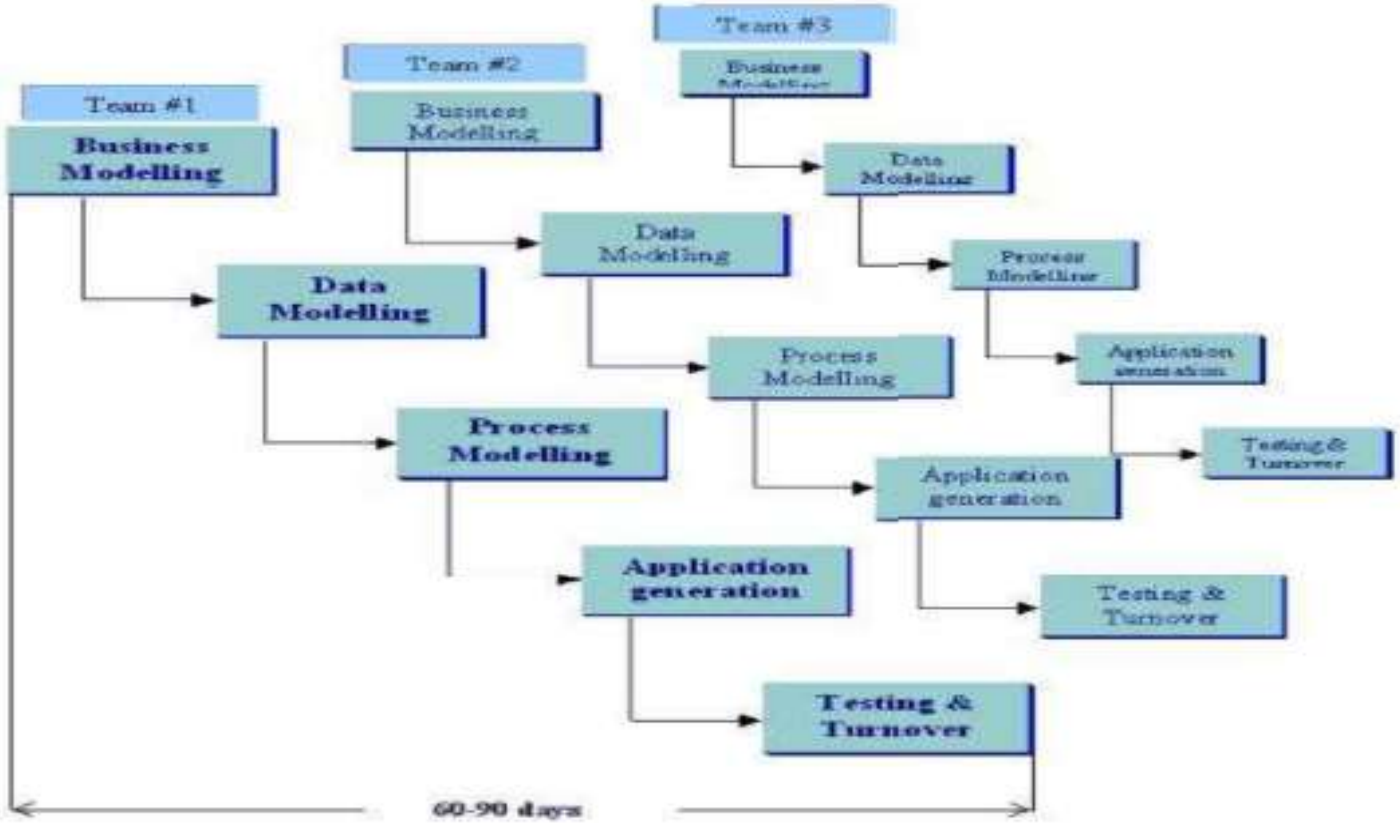
The spiral model is a **risk-driven** where the process is represented as spiral rather than a sequence of activities.

It was designed to include the best features from the waterfall and prototyping models, and introduces a new component; risk-assessment.



RAPID APPLICATION DEVELOPMENT MODEL

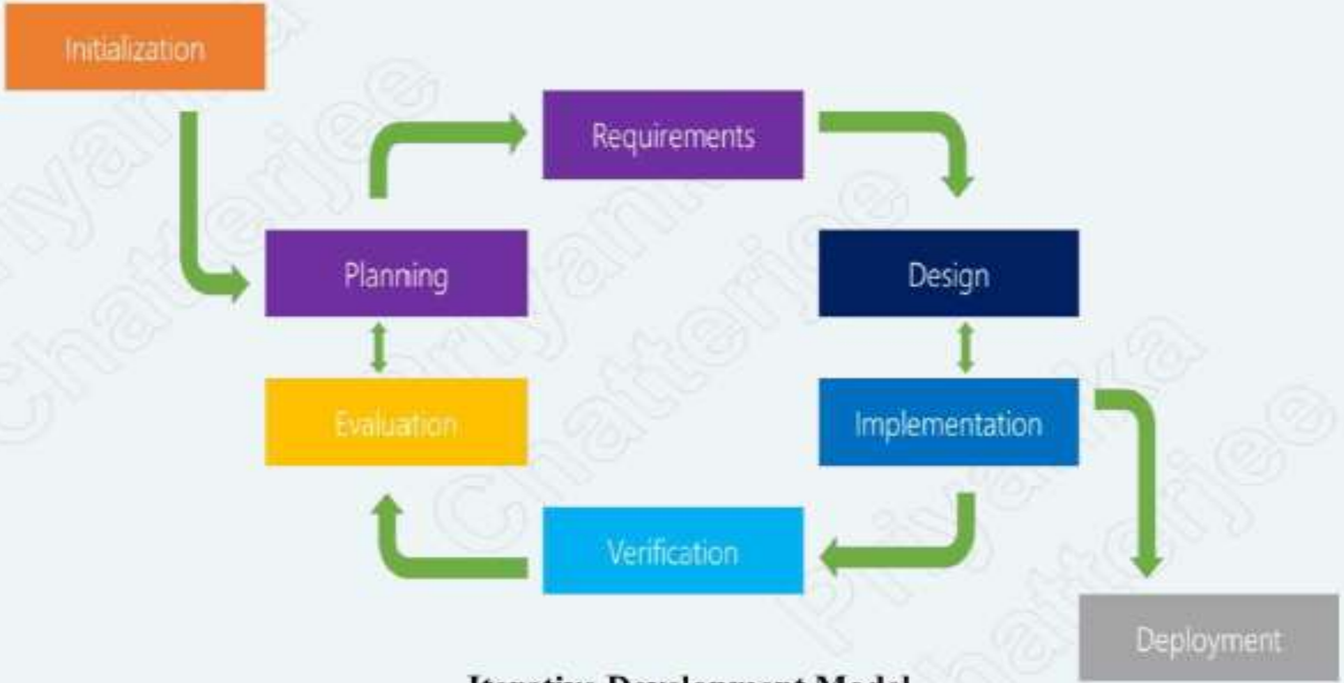
- Rapid Application Development(RAD) is an incremental software model that a short development cycle. The RAD model is a “high-speed” of the waterfall model. The RAD process enables a development team to create a fully functional system within a very short time period.



ITERATIVE DEVELOPMENT MODEL

Iterative development model aims to develop a system through building small portions of all the features, across all components.

We build a product which meets the initial scope and release it quickly for customer feedback. An early version with limited features important to establish market and get customer feedback.



Iterative Development Model

REQUIREMENT ENGINEERING

Requirements engineering (RE) refers to the process of defining, documenting, and maintaining requirements in the engineering design process. Requirement engineering provides the appropriate mechanism to understand what the customer desires, analyzing the need, and assessing feasibility, negotiating a reasonable solution, specifying the solution clearly, validating the specifications and managing the requirements as they are transformed into a working system.

It is a four-step process, which includes -

1. Feasibility Study

2. Requirement Elicitation and Analysis

3. Software Requirement Specification

4. Software Requirement Validation

5. Software Requirement Management

Steps of Requirement

1. Feasibility Study:

The objective behind the feasibility study is to create the reasons for developing the software that is acceptable to users, flexible to change and conformable to established standards.

2. Requirement Elicitation and Analysis:

This is also known as the gathering of requirements. Here, requirements are identified with the help of customers and existing systems processes, if available.

3. Software Requirement Specification:

Software requirement specification is a kind of document which is created by a software analyst after the requirements collected from the various sources - the requirement received by the customer written in ordinary language. It is the job of the analyst to write the requirement in technical language so that they can be understood by the development team.

Types of Requirements

System Requirements:

- System requirements set out the systems functions, services and operational constraints in detail. It may be part of the contract between the system buyer and the software developer.

Types of system requirements:

- Functional requirements.
- Non-functional Requirements.
- Domain Requirements.

Functional Requirements:

- The customer should provide statement of service. It should be clear how the system should react to particular inputs and how a particular system.

Types of Requirements(Contd.)

Non-Functional Requirements:

- The system properties and constraints various properties of a system can be: reliability, response time, storage requirements.

Domain Requirements:

Requirements can be application domain of the system, reflecting, characteristics of the domain.

User Requirements:

- User requirements are defined using natural language labels and diagrams because these are the representation that can be understood by all users.

DATA ENGINEERING

The main aim of data engineering is to generate a model which shows firmness, delight and commodity. Software design is an iterative process through which requirements are translated into the blueprint for building the software.

The set of fundamental software design concepts are as follows:

1. Abstraction

A solution is stated in large terms using the language of the problem environment at the highest level abstraction. The lower level of abstraction provides a more detail description of the solution.

2. Architecture

The complete structure of the software is known as software architecture. The architecture is the structure of program modules where they interact with each other in a specialized way.

3. Modularity

Software is separately divided into name and addressable components. Sometime they are called as modules which integrate to satisfy the problem requirements. Modularity is the single attribute of a software that permits a program to be managed easily.

Software Design Concepts

4.Information hiding

Modules must be specified and designed so that the information like algorithm and data presented in a module is not accessible for other modules not requiring that information.

5.Cohesion

Cohesion is an extension of the information hiding concept.

A cohesive module performs a single task and it requires a small interaction with the other components in other parts of the program.

6.Coupling

Coupling is an indication of interconnection between modules in a structure of software

7. Refinement

Refinement is a top-down design approach. It is a process of elaboration. A program is established for refining levels of procedural details.

A hierarchy is established by decomposing a statement of function in a stepwise manner till the programming language statement are reached.

Software Design Process

The design phase of software development deals with transforming the customer requirements as described in the SRS documents into a form implementable using a programming language.

The software design process can be divided into the following three levels of phases of design:

1.Interface Design

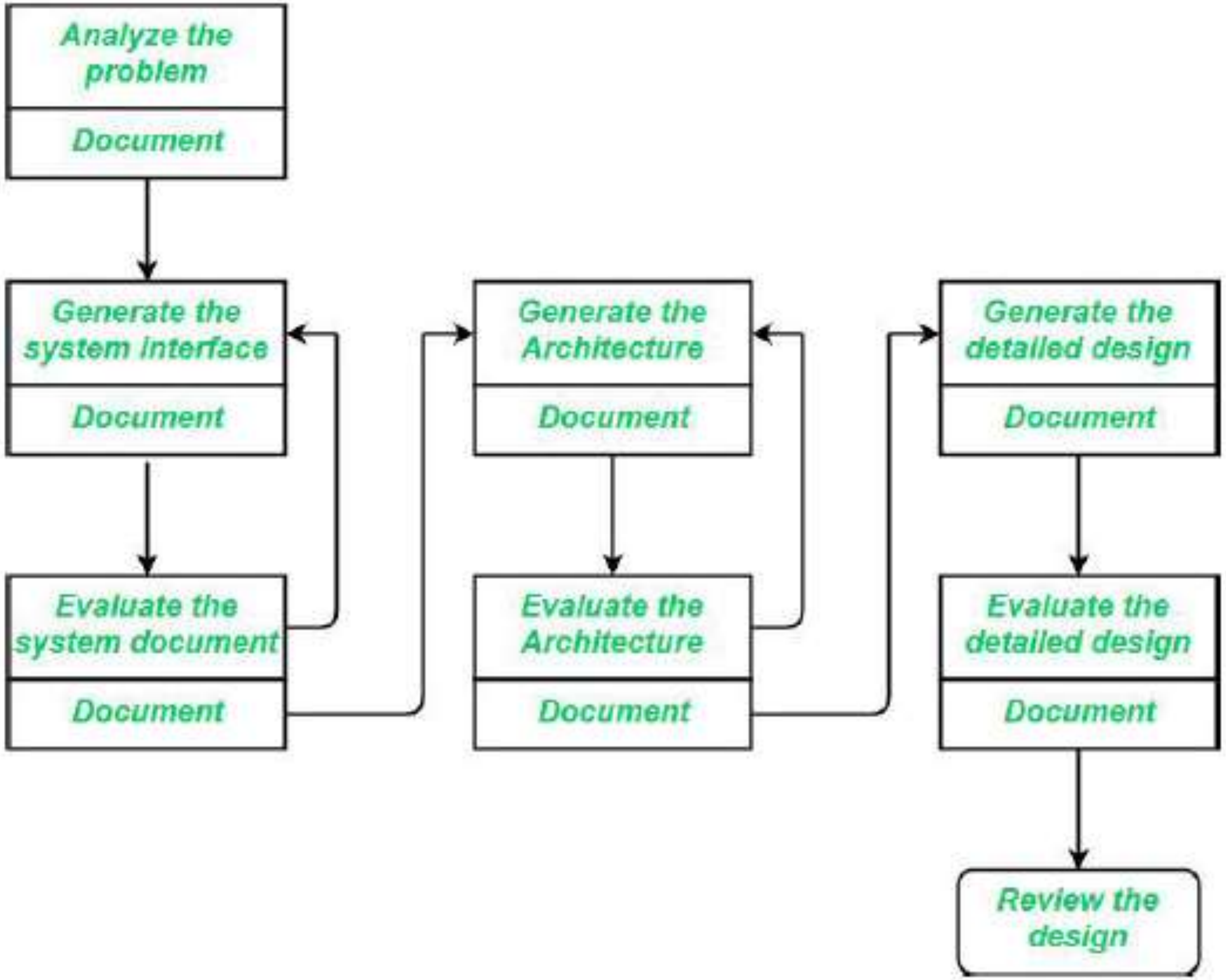
2. Architectural Design

3. Detailed Design

1.Interface Design:

Interface design is the specification of the interaction between a system and its environment. this phase proceeds at a high level of abstraction with respect to the inner workings of the system i.e, during interface design, the internal of the systems are completely ignored and the system is treated as a black box.

Software Design Process(Contd.)


Analyze the problem
Document
Generate the system interface
Document
Evaluate the system document
Document
Generate the Architecture
Document
Evaluate the Architecture
Document
Generate the detailed design
Document
Evaluate the detailed design
Document
Review the design

Software Design Process(Contd.)

Architectural Design:

The software needs the architectural design to represents the design of software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectural styles.

Each style will describe a system category that consists of :

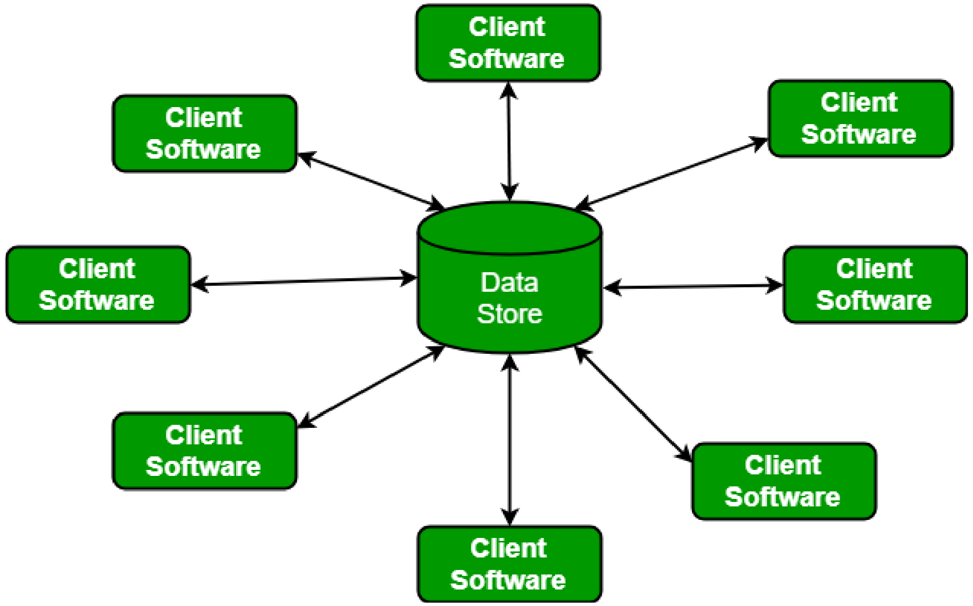
A set of components (eg: a database, computational modules) that will perform a function required by the system. The set of connectors will help in coordination, communication, and cooperation between the components. Conditions that how components can be integrated to form the system.

Semantic models that help the designer to understand the overall properties of the system. The use of architectural styles is to establish a structure for all the components of the system.

Taxonomy of Architectural Styles

1.Data centered architectures:

A data store will reside at the center of this architecture and is accessed frequently by the other components that update, add, delete or modify the data present within the store.

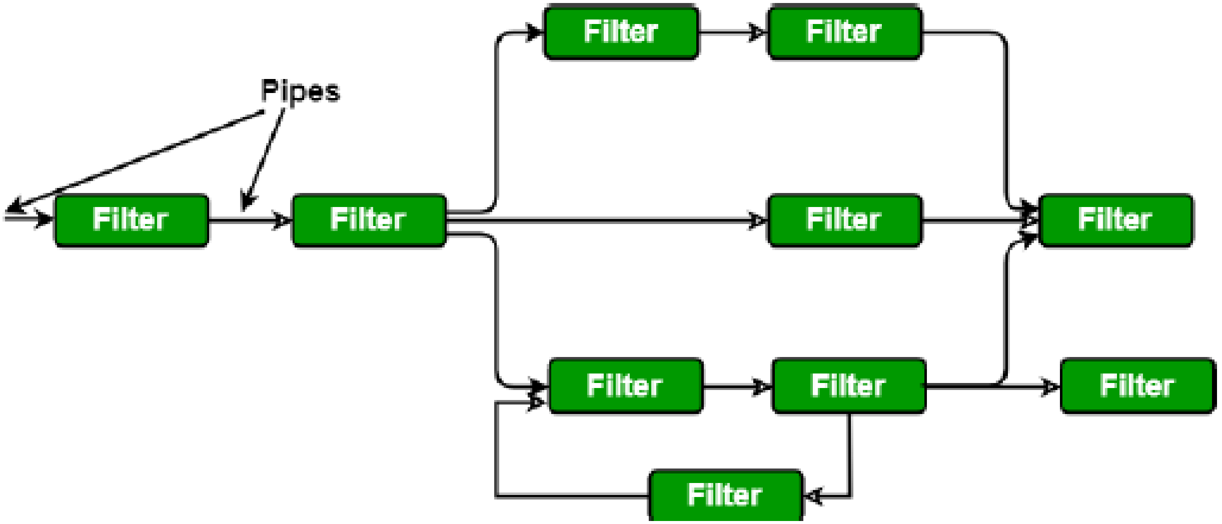


The data-centered architecture will promote integrability. This means that the existing components can be changed and new client components can be added to the architecture without the permission or concern of other clients.

Taxonomy of Architectural Styles(Contd.)

Data flow architectures

This kind of architecture is used when input data to be transformed into output data through a series of computational manipulative components.



Pipes are used to transmit data from one component to the next. Each **filter** will work independently and is designed to take data input of a certain form and produces data output to the next filter of a specified form. The filters don't require any knowledge of the working of neighboring filters.

Software Testing

Software testing can be stated as the process of verifying and validating that a software or application is bug free, meets the technical requirements as guided by it's design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases.

Software testing can be divided into two steps:

1. **Verification**: it refers to the set of tasks that ensure that software correctly implements a specific function.
2. **Validation:** it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

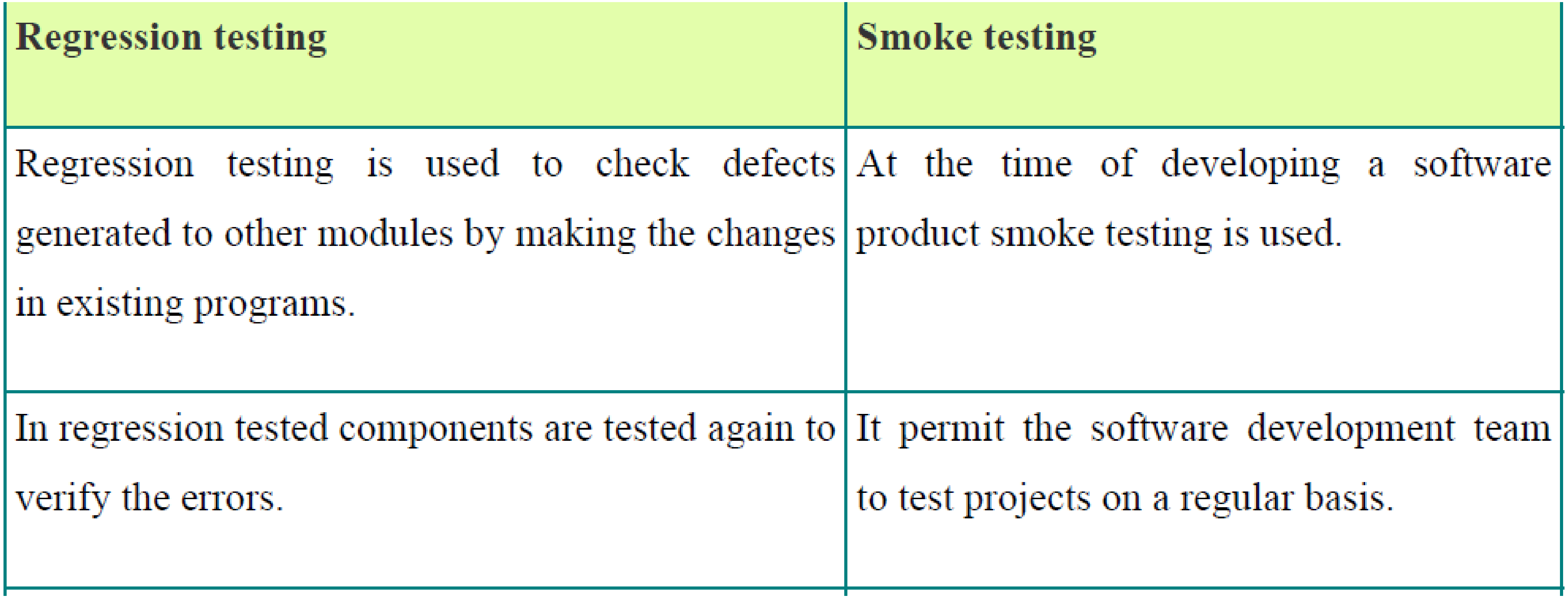
Software Testing: Types

1. **Unit Testing:** A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.
2. **Integration Testing:** A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.
3. **System Testing:** A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.
4. **Acceptance Testing:** A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Software Testing: Types(Contd.)

Regression testing: In regression testing the software architecture changes every time when a new module is added as part of integration testing.

Smoke testing: The developed software component are translated into code and merge to complete the product.



Regression testing	**Smoke testing**
Regression testing is used to check defects generated to other modules by making the changes in existing programs.	At the time of developing a software product smoke testing is used.
In regression tested components are tested again to verify the errors.	It permit the software development team to test projects on a regular basis.

System Testing: Types

Performance Testing:

Performance Testing is a type of software testing that is carried out to test the speed,scalability, stability and reliability of the software product or application.

Load Testing:

Load Testing is a type of software Testing which is carried out to determine the behavior of a system or software product under extreme load.

Stress Testing:

Stress Testing is a type of software testing performed to check the robustness of the system under the varying loads.

Scalability Testing:

Scalability Testing is a type of software testing which is carried out to check the performance of a software application or system in terms of its capability to scale up or scale down the number of user request load.

Black-box and White-box Testing

1. Black Box Testing: The technique of testing in which the tester doesn’t have access to the source code of the software and is conducted at the software interface without concerning with the internal logical structure of the software is known as black box testing.

2. White-Box Testing: The technique of testing in which the tester is aware of the internal workings of the product, have access to it’s source code and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.

Control Structure Testing

Control structure testing is used to increase the coverage area by testing various control structures present in the program. The different types of testing performed under control structure testing are as follows-

1. Condition Testing

2. Data Flow Testing

3. Loop Testing

Project Management

The Management Spectrum:

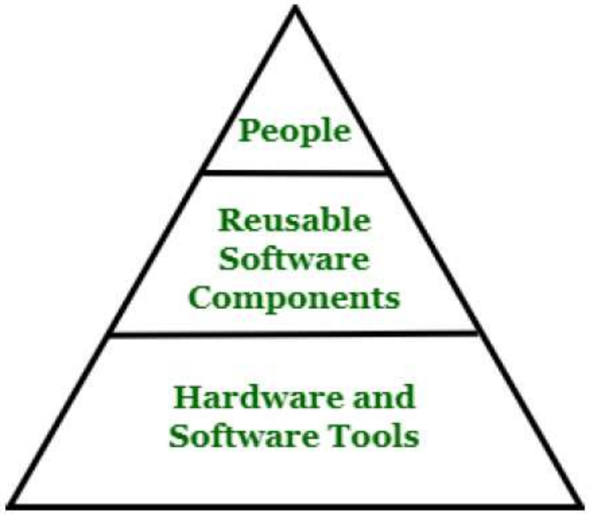
In software engineering, the management spectrum describes the management of a software project. The management of a software project starts from requirement analysis and finishes based on the nature of the product, it may or may not end because almost all software products faces changes and requires support. It is about turning the project from plan to reality.

The four P's of management spectrum are

1. People
2. Product
3. Process
4. Project

Project Resources

Project resources simply mean resources that are required for successful development and completion of project. These resources can be capital, people, material, tool, or supplies that are helpful to carry out certain tasks in project. Without these resources, it is impossible to complete project.



Resource Pyramid

There are three types of resources that are considered and are very essential for execution of project and completion of project on time and on budget.

Types of Project Resources

1. Human Resource

Human plays an important role in software development process. No matter what size is and how much complexity is there in project, if you want to perform project task in an effective manner, then human resources are very essential.

2.Reusable Components

For bringing ease in software development process or to accelerate development process software, industry prefers to use some ready software components. The components can be defined as the software building blocks that can be created .

3.Hardware and Software tools

These are actually material resources that are part of project. This type of resource should be planned before starting development of project otherwise it way causes problems for the project.

Software Quality Assurance(SQA)

Software Quality Assurance (SQA) is simply a way to assure quality in the software. It

is the set of activities which ensure processes, procedures as well as standards suitable for the

project and implemented correctly.

Major Software Quality Assurance Activities:

1. SQA Management Plan:

Make a plan how you will carry out the sqa through out the project. Think which set of software engineering activities are the best for project.check level of sqa team skills.

2. Set The Check Points:

SQA team should set checkpoints. Evaluate the performance of the project on the basis of

collected data on different check points.

3. Multi testing Strategy:

Do not depend on single testing approach. When you have lot of testing approaches available use them.

Formal Technical Review

Formal Technical review is a software quality assurance activity performed by software engineer.

Objectives of FTR

1. FTR is useful to uncover error in logic, function and implementation for any representation of the software.

2. The purpose of FTR is to ensure that software meets specified requirements.

3. It is also ensure that software is represented according to predefined standards.

4. It helps to review the uniformity in software development process.

5. It makes the project more manageable.

Besides the above mentioned objectives, the purpose of FTR is to enable junior engineer to observer the analysis, design, coding and testing approach more closely.