NETAJI SUBHAS UNIVERSITY

Subject Name:- Software Engineering

Faculty Name:- Shadma Nigar

DEPARTMENT NAME:- IT Department

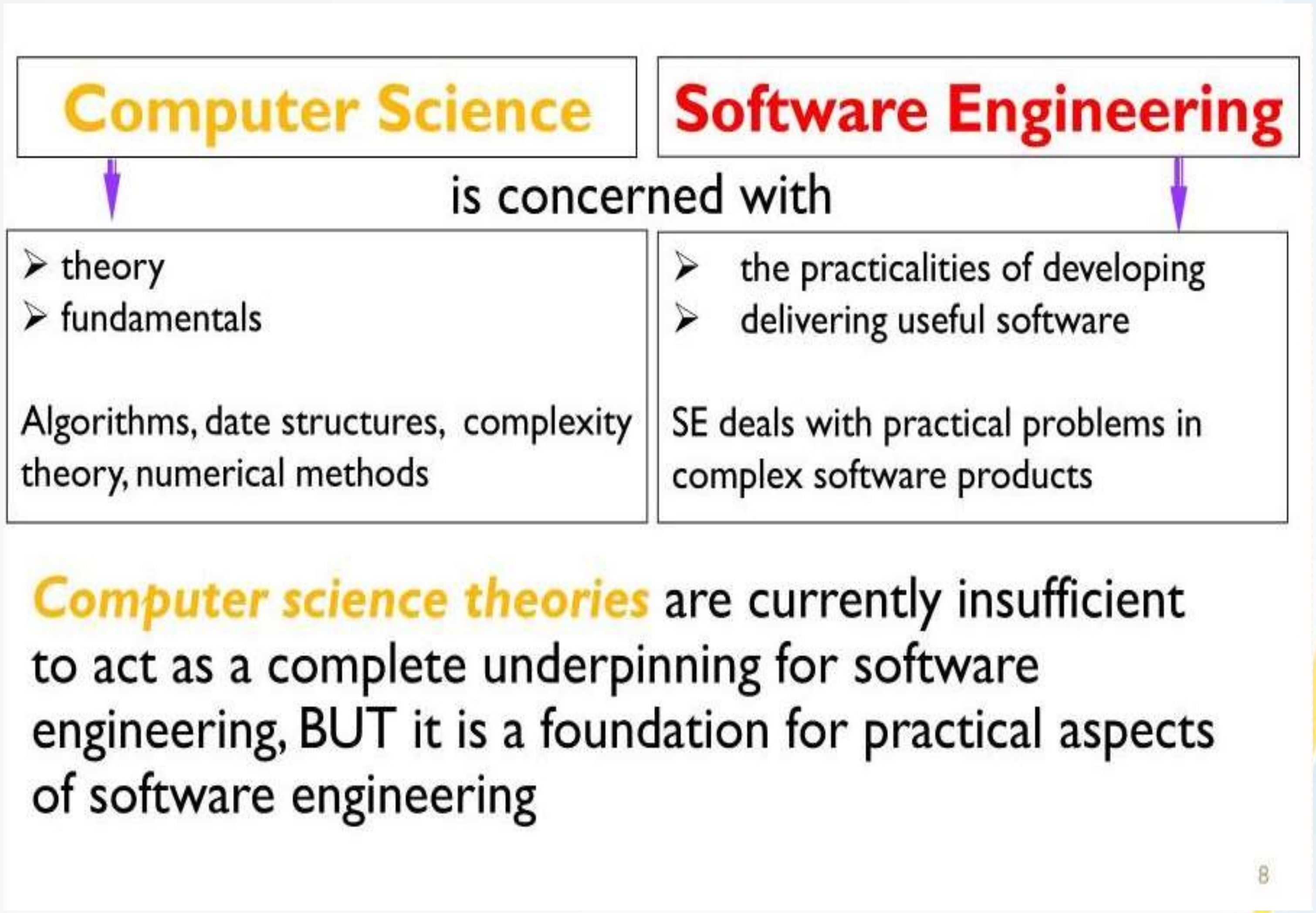
INTRODUCTION:

- Software is more than just a program code.
- A program is an executable code, which servers some computational purpose.
- Software is the collection of computer programs, procedures rules and associated documentation and data.
- Software is an information transformer- producing, managing, modifying, displaying or transforming information that can simple as a single bit or a complex as a multimedia application.

Software Products:

- Software products may be developed for a particular customer or may be developed for a general market.
- **Software products may be:**
    - Generic
    - Bespoke
- **What are the attributes of good software?**
    - Maintainability.
    - Dependability
    - Efficiency
    - Usability

What is the difference between software engineering and computer science?



Computer Science | Software Engineering

is concerned with

- theory
- fundamentals

Algorithms, date structures, complexity theory, numerical methods

- the practicalities of developing
- delivering useful software

SE deals with practical problems in complex software products

Computer science theories are currently insufficient to act as a complete underpinning for software engineering, BUT it is a foundation for practical aspects of software engineering

8

Software Engineering Paradigms:

Software Characteristics:

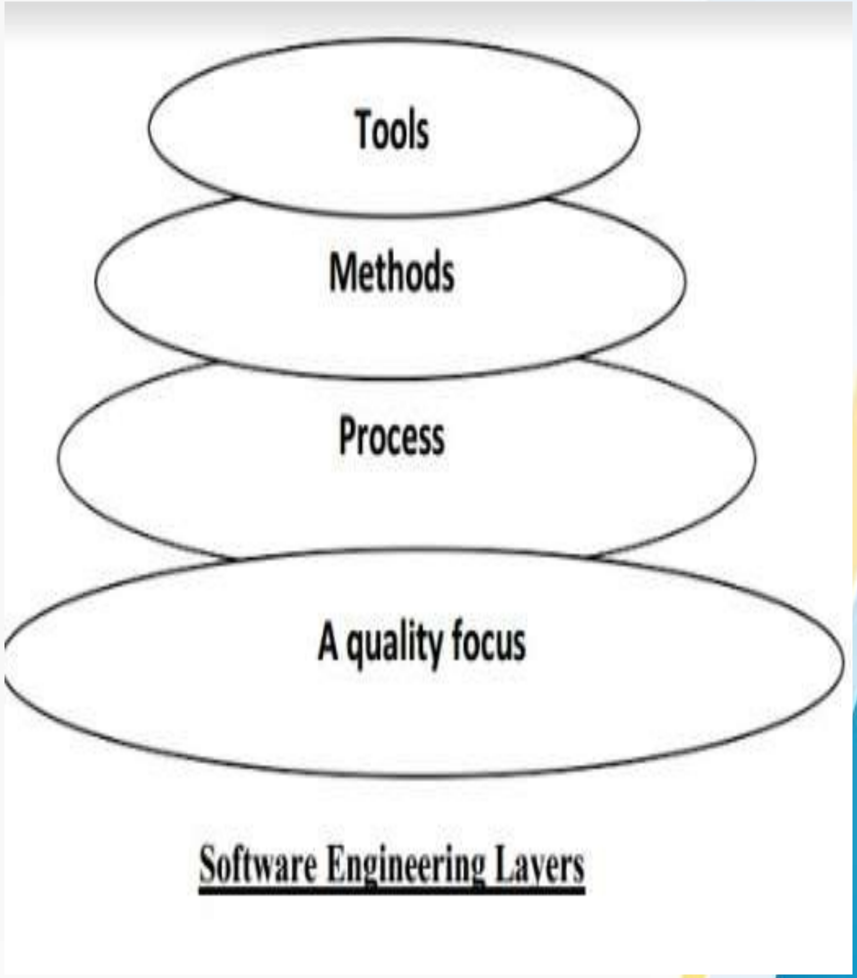
- Software is developed or engineered, it is not manufactured in the classical sence.
- Software doesn’t “wear out”.
- Although the industry is moving towards component based assembly, most software continues to be custom to built.

Software Applications Types:

- System Software.
- Real-time Software.
- Business Software.
- Engineering and Scientific Software.
- Embedded Software.
- Personal Computer Software.
- Web-based Software.
- Artifical Intelligence Software.

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

What are the five generic process framework activities?

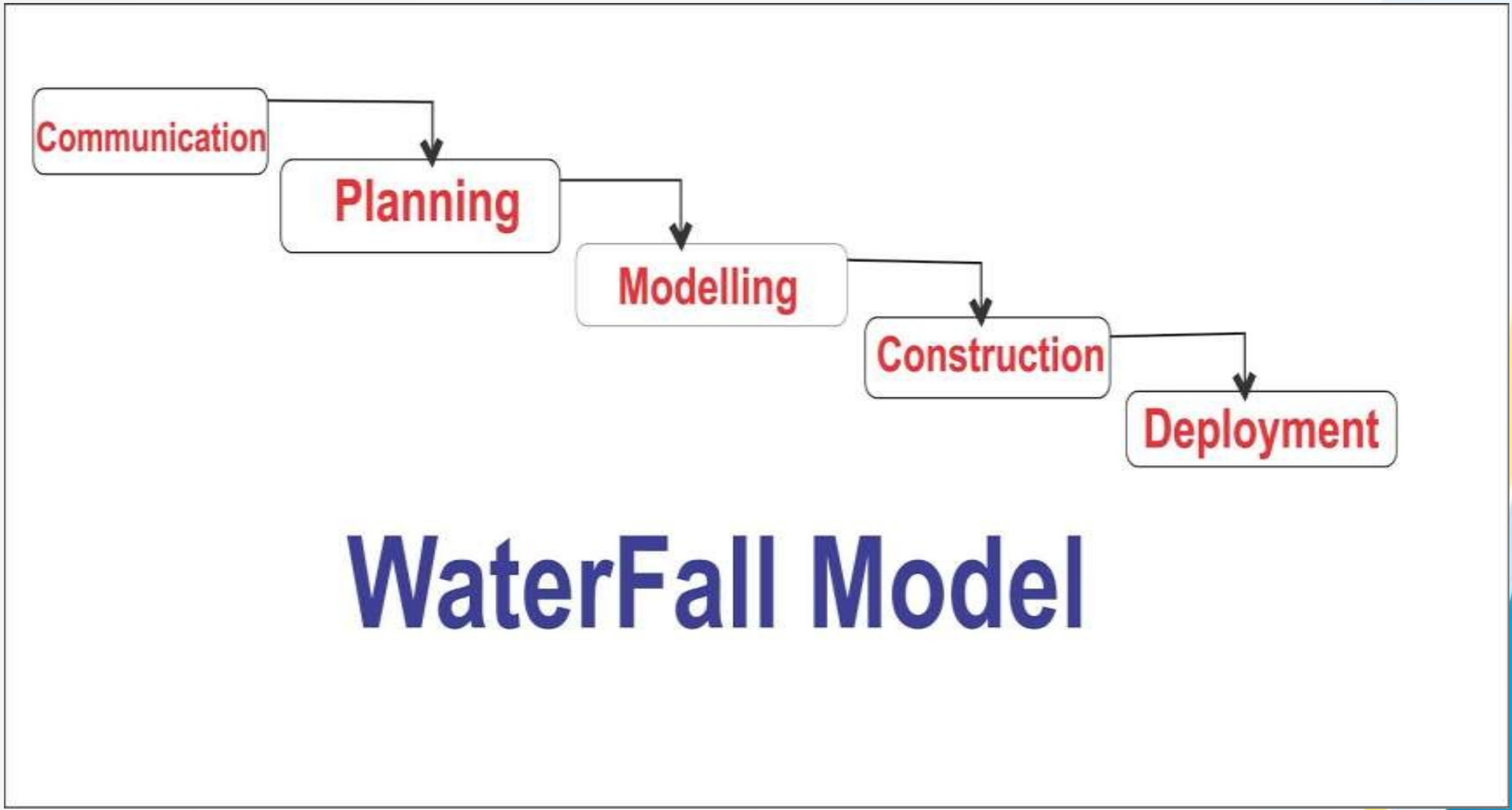
- The following generic process framework is applicable to the majority of software projects.
  - Communication.
  - Planning.
  - Modeling.
  - Construction.
  - Deployment.

Process Models:

- Every software engineering organization should describe a unique set of framework activities for the software process it adopts.
  - Waterfall Life Cycle Model.
  - Iterative Waterfall Life Cycle Model.
  - Prototyping Model.
  - Incremental Model.
  - Sprial Model.
  - RAD Model.
  - Sprial Model.

Waterfall Life Cycle Model.

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


Communication
Planning
Modelling
Construction
Deployment
WaterFall Model

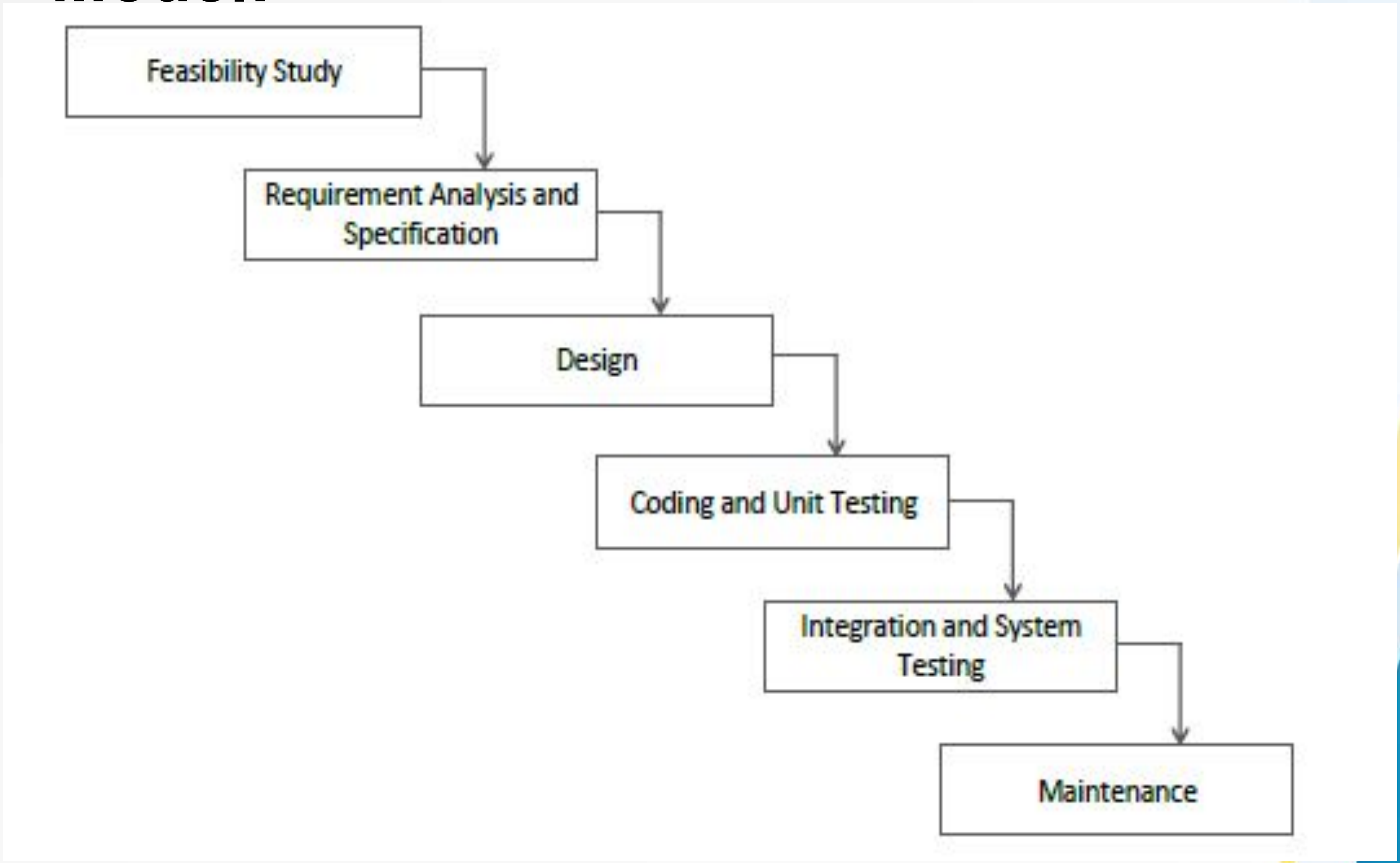
Advantages:

- Easy to understand.
- Each phase has well defined input and output.
- Helps project manger in proper planning of project.
- Provides a templates into which methods of analysis, design, code and support can be placed.

Disadvantages:

- One way street.
- It lack overlapping and interactions among phases.
- Model doesn’t support delivery of system in pieces.

Phases of the Classical Waterfall Model:


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

Feasibility Study:

- It involves analysis of the problem and collection of allrelevant information relating to the product.
- The collected data are analysed.
  - Requriments of the Customer.
  - Formulations of the different strategies for solving the problem.
  - Evaluation of different solution strategies.

Requriments Analysis and Specification:

- It is understand the exact requriments of the customer and to document them properly.
  - Requirements gathering and analysis.
  - Requirements specification.

Design:

- The deign phase is to transform the requirements specified in the document into a structure that is suitable for implementation in some programming languaage.
    - Traditional Design Approach.
    - Object-Oriented Design Approach.

Coding and Unit Testing:

- The purpose mof the coding and unit testing phase of software development is to translate the software design into source code.

Integration and System Testing:

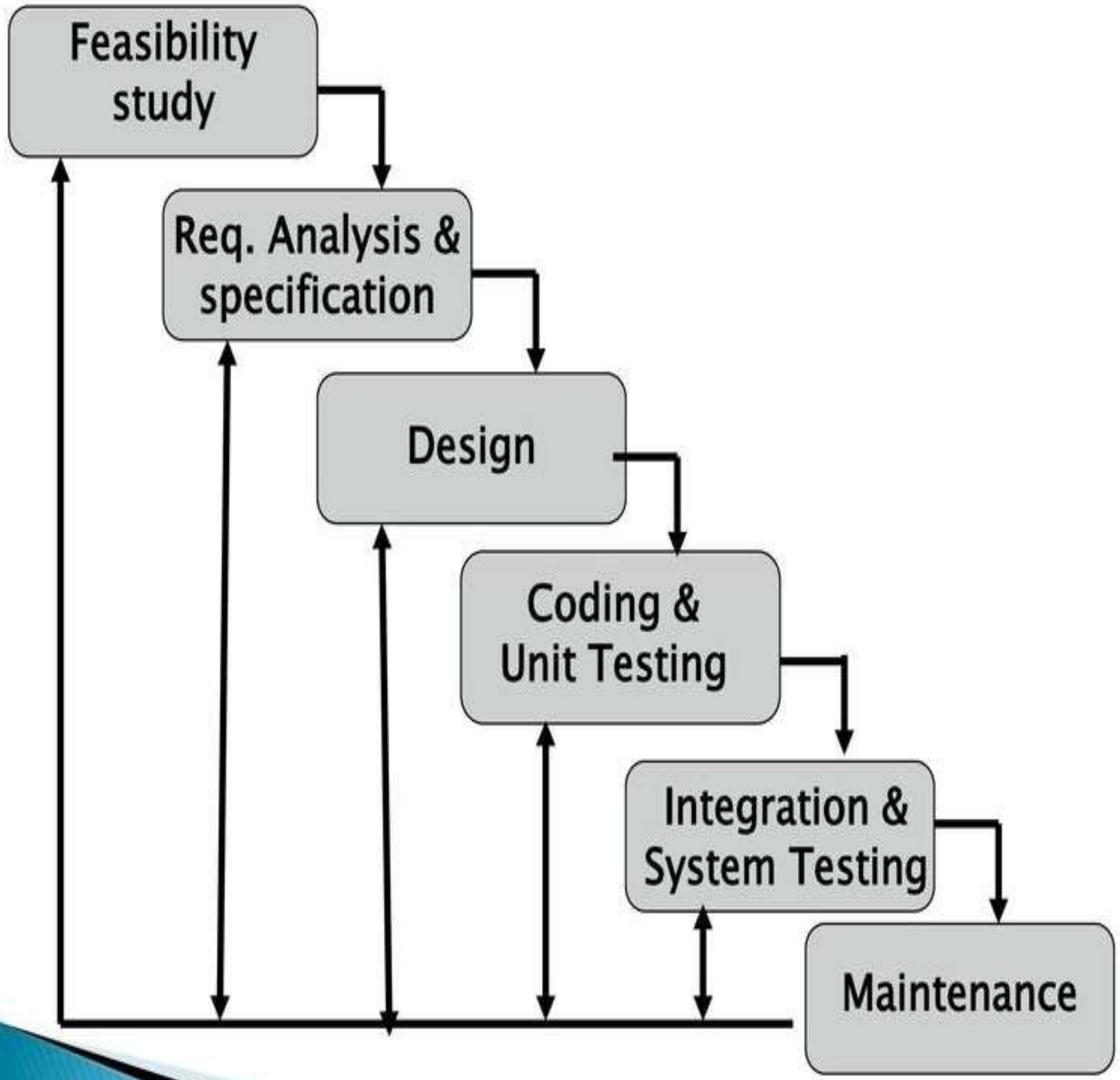
- ‘Integration of different modules is coded and unit tested.
    - $\alpha - \quad Testing$
    - $\beta - \quad Testing$
    - Accsptance Testing.

Maintenance:

- Maintenance of a typical software products requires much more than the effort necessary to develope the product itself.

Iterative Waterfall life cycle model:

- The main changes is done by providing feedback paths from every phase to its preceding phase.


Feasibility study
Req. Analysis & specification
Design
Coding & Unit Testing
Integration & System Testing
Maintenance

Prototype Model:

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

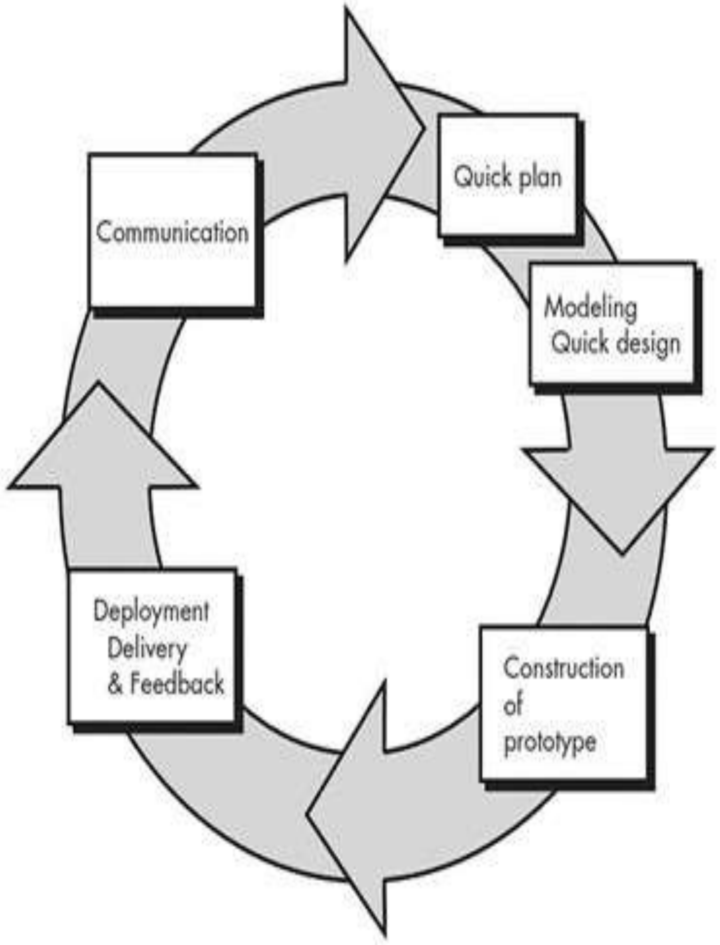


Figure: Prototype Model

Advantages:

- Clarity.
- Risk Identification.
- Good Environment.
- Take less time to complete.

Disadvantages:

- High cost.
- Slow process.
- Too many changes.

RAD 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.

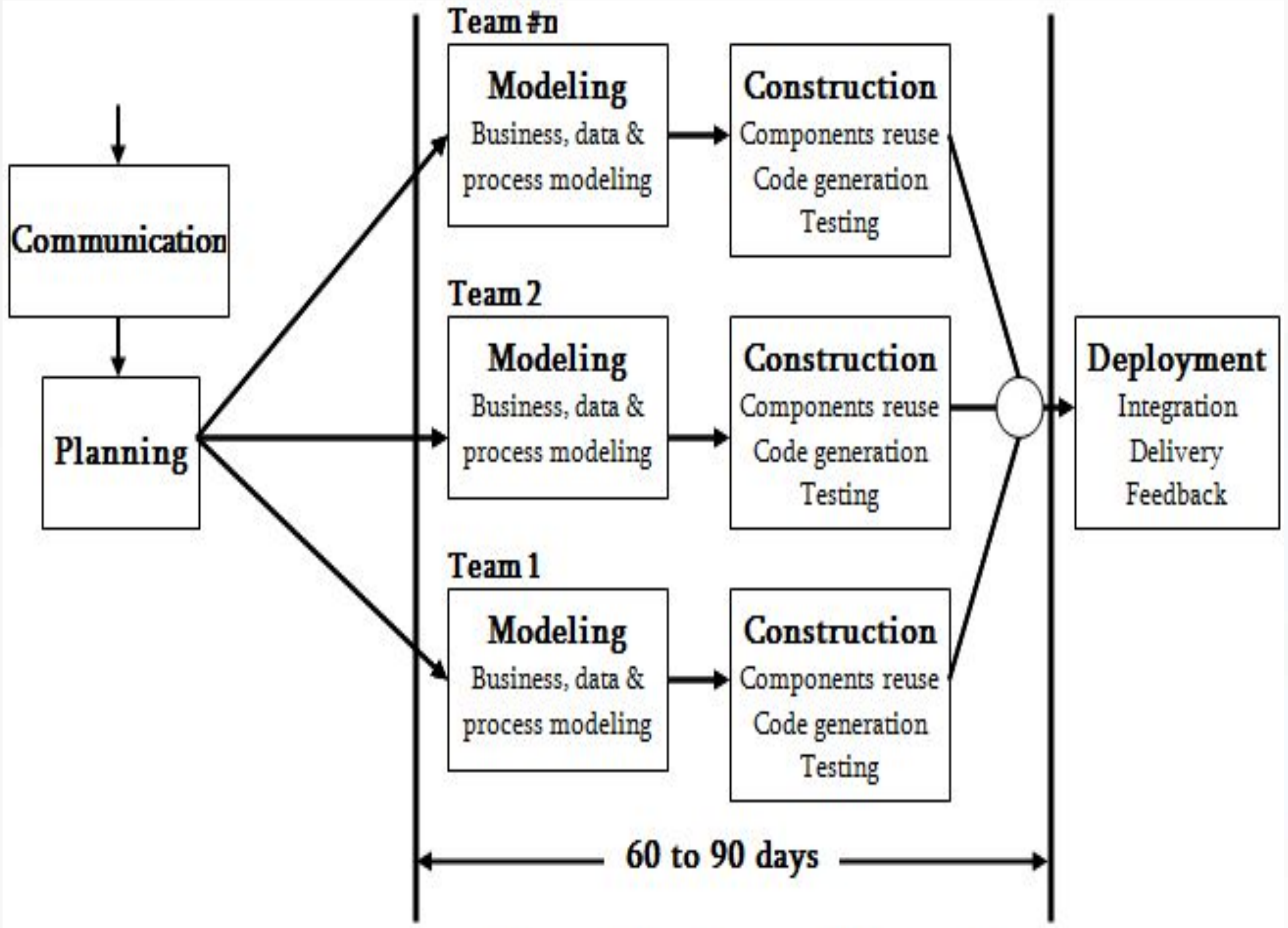


Figure : Flowchart of RAD model

Contents of RAD Pakages:

- Graphical user development environment.
- Reusable Components.
- Code generator.
- Programming Language.

Advantages:

- Fast products.
- Efficient Documentation.
- Interaction with user.

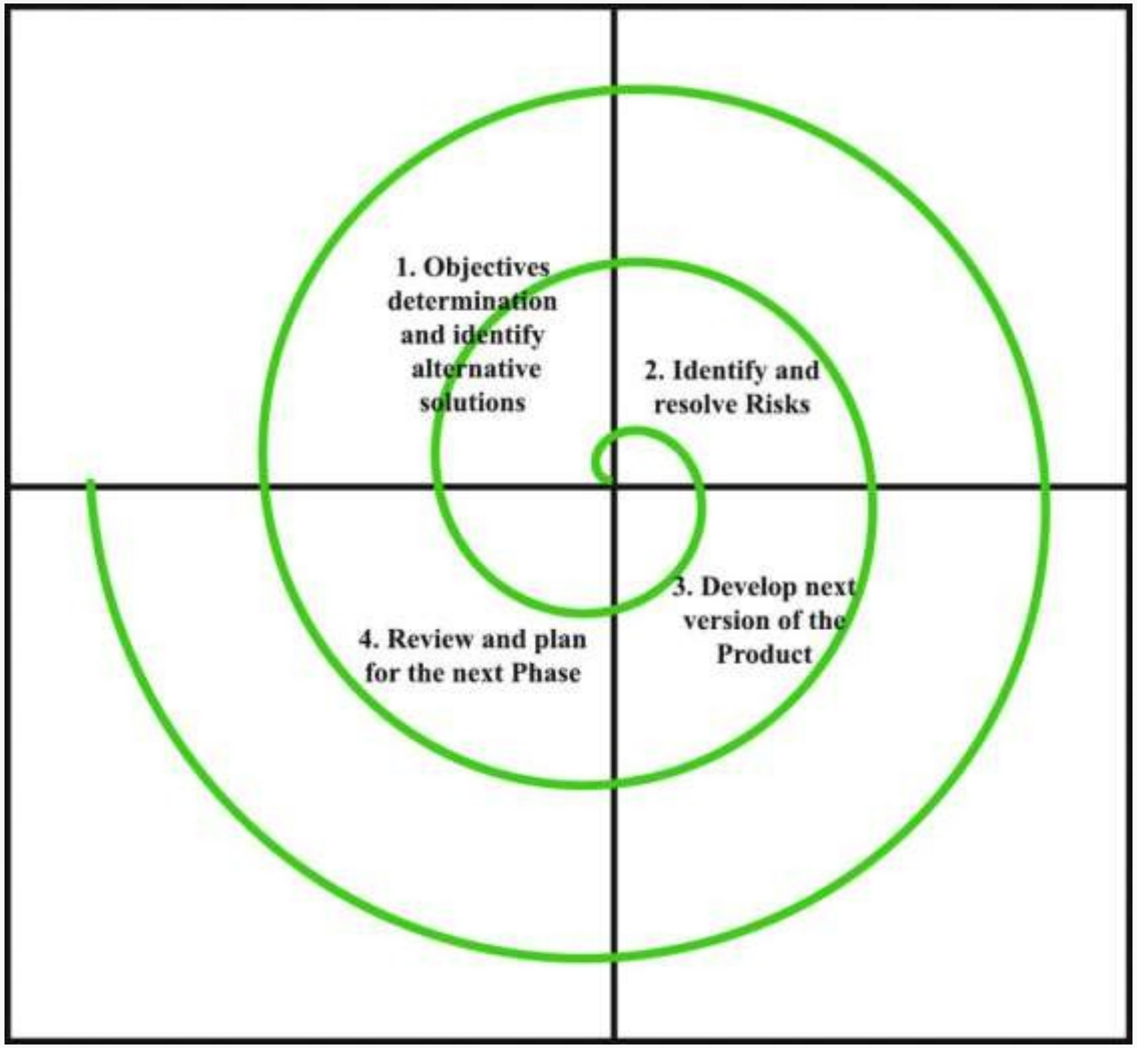
Disadvantages:

- User may not like fast activities.
- Not suitable for technical risks.

Sprial Model :

- This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis.
- The spiral model has four phases: Planning, Design, Construct and Evaluation.

Quadrants in sprial model :

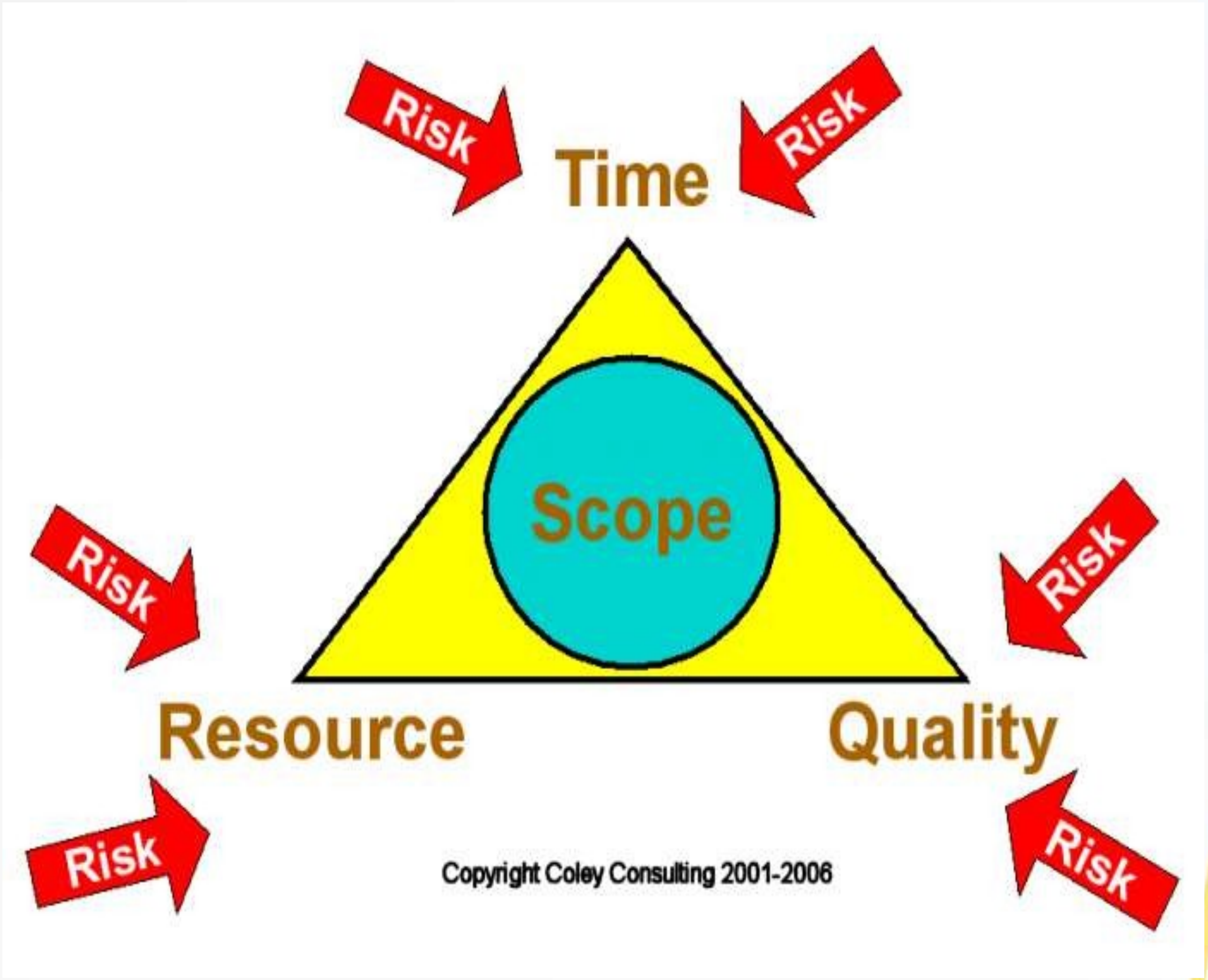

1. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase

Advantages :

- Risk Identification at early stage.
- Suitable for high rk projects.
- Flexibility for adding functionaility.

Disadvantages:

- Costly.
- Risk dependent.
- Not suitable for smaller projects.
- Difficult to meeting budget.


Risk
Risk
Time
Scope
Risk
Risk
Resource
Quality
Risk
Risk
Copyright Coley Consulting 2001-2006

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.

Problem of Functional Requirements:

- User Intention.
- Developer Interpretation.
- Requirements completness and consistency:

Non-Functional Requirements:

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

Types of Non-Functional Requirements:

- Product Requirements.
- Organizational Requirements.
- External Requirements.

Domain Requirements:

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

Problem of Domain Requements:

- Understandability.
- Implicitness.

User Requirements:

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

- Client Managers.
- System End Users.
- Client Engineers.
- Contract Managers.

Problem of User Requirements:

- Lack of Clarity.
- Requirements Confusion.
- Requirements Mixture.

Software Requirement Specification:

- Software Requirements document is the specification of the system.
- It is not a design document.
- Requirements document is called SRS.

Users of SRS:

- Users, Customer and marketing Personnel.
- Software Developers.
- Test Engineers.
- Project Managers.
- Maintenance Engineers.