ROHINI
COLLEGE OF ENGINEERING AND TECHNOLOGY
Approved by AICTE and affiliated to Anna University,(An ISO Certified Institution)
Accredited by NAAC with A+ Grade

Recognized Under Section 2(f) of University Grants Commission, UGC Act 1956

Department of Management Studies

MBA – I Semester

BA4106 Information Management

Dr.Jackson Daniel
Professor/ECE Department

UNIT –II

System Development Methodologies

1. System Development Methodologies

“System development Methodology refers to the framework that is used to structure, plan and control the process of developing an information system.”

“a standard process followed by an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems"

“ Body of methods, rules and postulates used by system practitioners to investigate, understand, and address systems, their issues, problems, behaviors, and context and where appropriate – to moderate, modify, or otherwise solve, resolve or dissolve issues and problems”

1. System Development Methodologies Cont'd

Objectives of System Methodology

- How the system should be **working** ?
- To understand the **relationship** among various activities
- A well-designed system can help ensure that the system is **reliable**, **efficient**, and **user-friendly**.
- **Take into account any constraints or limitations**
- **Completeness**: System design should meet all user requirements
- Can also reveal **data collection needs.**

1. System Development Methodologies Cont'd…

Different Types of System Development Methodology

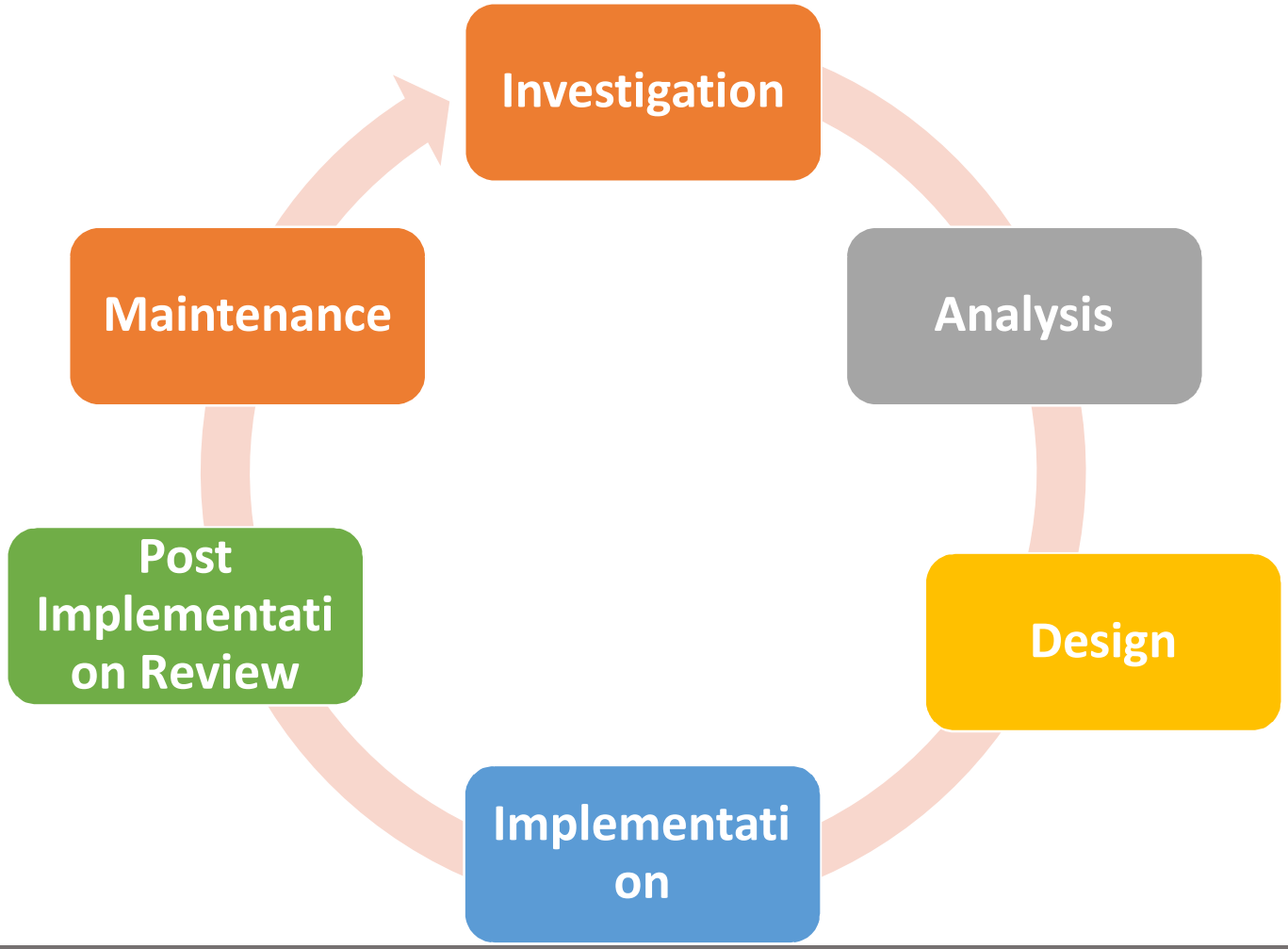
- System Development Life Cycle (SDLC)
- Waterfall Model
- Prototyping Model
- Spiral Model
- Rapid Application Development (RAD)

System Development Life Cycle (SDLC)

- SDLC is used by **analysts** to develop an information system
- The System Development Life Cycle (SDLC) provides a **well-structured framework** that gives an idea, of how to build a system.
- SDLC can apply to **technical and non-technical systems**.
- it defines the necessary steps needed to take a project from the **idea** or concept stage to the actual **deployment** and further **maintenance**.

System Development Life Cycle (SDLC)

Steps of SDLC



Steps of SDLC

Investigation Phase:

- ❑ Does there exist any business problem?
- ❑ What is the reason for problem?
- ❑ Will new information system be able to solve the problem
- ❑ What can be feasible IS solution?
- ❑ It includes following steps
  - ➢ System Planning and Selection
  - ➢ Feasibility Analysis
  - ➢ Feasibility Report

Steps of SDLC

Investigation Phase:

System Planning and Selection

- **Identifying** and Selecting Projects/proposals
- identifying project **activities**
- The Process of **Initiating** and Planning Systems Development Projects
- Deliverables and **Outcomes** – Schedule of IS development project

Feasibility Analysis

- Analyze-project will be a success or not
- Assessment of potential impact of project
- Different types of feasibility study
  - Technical Feasibility
  - Economic Feasibility
  - Political Feasibility
  - Legal Feasibility
  - Behavioral Feasibility

Feasibility Report

- Covering Letter
- Table of Contents
- Overview of the study
- Detailed findings
- Recommendations and conclusions
- System Design and implementation schedule
- Appendix (documents/data compiled during the study)

System Analysis:

- Gathering and understanding the requirements of the system.
- Conducting interviews, studying existing processes, and identifying stakeholders' needs
- To understand " What to do?" and "how to do?" – get solution.
- Furthermore, developers will often create a software requirement specification
- Generate the system requirements
- Evaluate existing prototypes.
- Conduct market research.
- Set concrete goals.
- It is user controlled stage

Developers should do these

3. System Design:

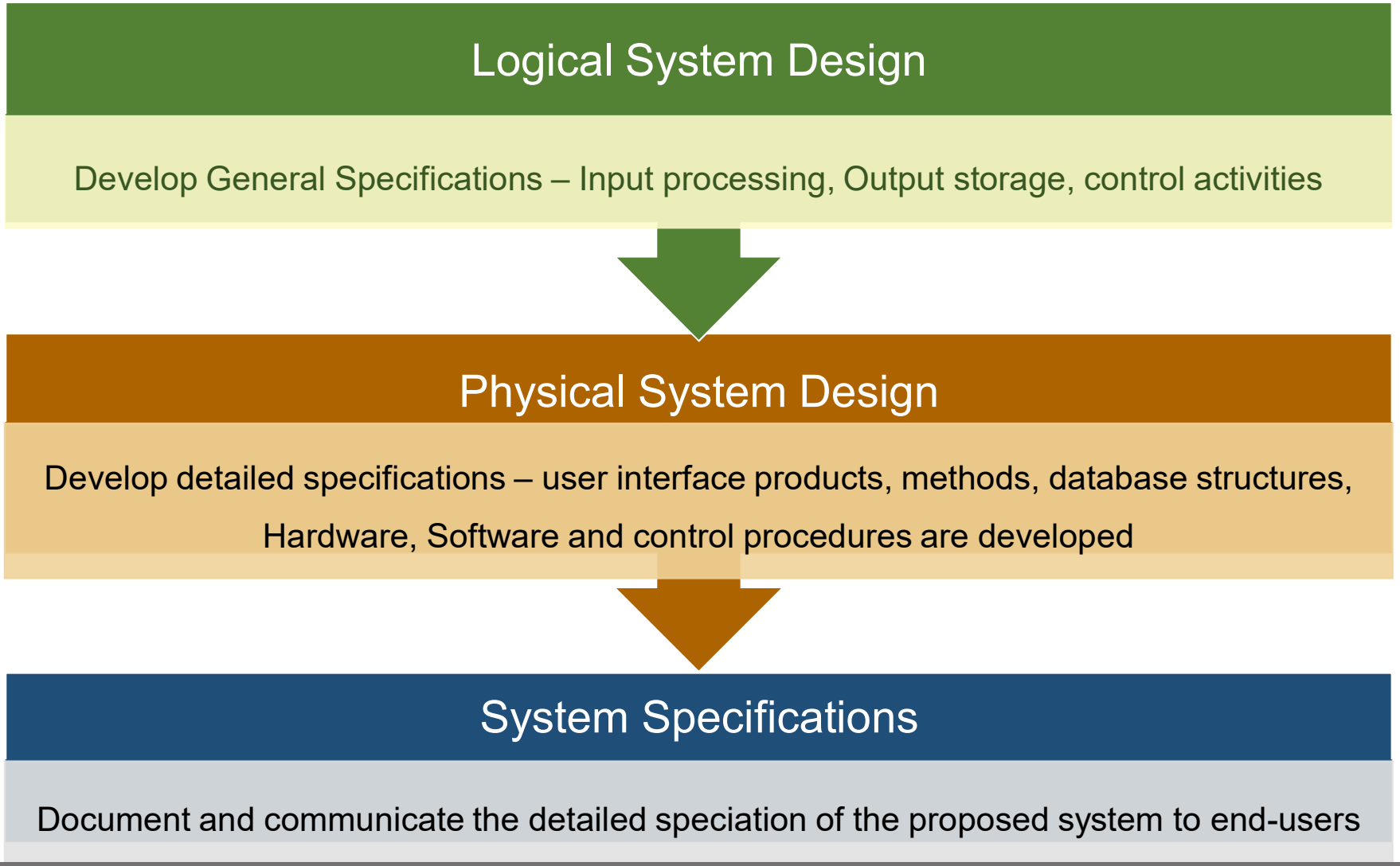
- System design identifies " how" the system will accomplish the change.

- An analyst engages in the following action during system design phase.
  - Plans design activities
  - Works with user to decide different data input to system
  - Draw models for the new system using data flow diagram and ERD
  - Clearly describe the data requirements
  - Write down the program specifications
  - Recognizes and orders hardware and software whenever required.

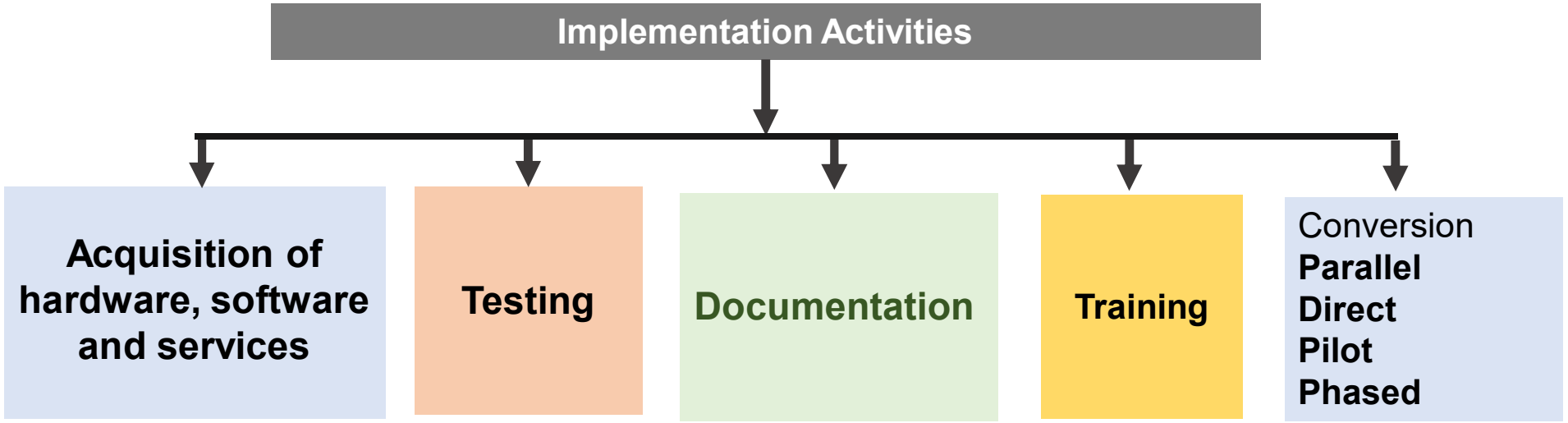
- System design includes both logical and physical design

System Design:

Activities of System Design



4. Implementation



- Implementation **is time consuming** and **complex process**.
- If Implementation is **not properly done**, the designed system will fail.
- Implement the design into **source code** through coding.
- **Combine** all the modules together into training environment that detects errors and defects.
- **Integrate** the information system into its environment and **install** the new system.

Acquisition of hardware, software and services

- System specifications are **planned** during the design stage
- **Call suppliers** to present proposals
- Establish **minimum acceptable** physical and performance characteristics (S/W & H/W)
- **Assign score** to various proposals
- This shows strengths and weaknesses of every proposal.
- Large companies, Process the special “benchmark test programs” and test the proposal.
- Test results are evaluated by user.
- Thus acquisition process involves selection of H/W, S/W

Testing

- Once the software is complete, and it is deployed in the testing environment.
- Set of input, “test cases” are given to the program
- Programmers observe the program – correction / debugging
- Ensuring Quality of the system.

Unit Testing

- Code into consideration.
- Components are tested one by one
- Each component is independently Tested

Integration Testing

- testing the interface between two software units or modules..
- Its focus is on determining the correctness of the interface.

Validation Testing

- Testing after the system is assembled as single unit
- Ensures - business and end-user requirements.
- It is a validation of the actual and expected products

System Testing

- Evaluates the overall functionality and performance of a complete and fully integrated software solution.
- System Testing is a black-box testing.

Documentation

- SDLC documentation is the process of creating, organizing, and updating various types of documents that describe the different phases, activities, and deliverables of a software development project.
- The support can be inbuilt within the system or made available online and can take many forms such as
  - System messages
  - Online help facilities
  - Online tutorials
  - Operation manuals
  - User’s guides
  - Information centers
  - Embedded training

The following purposes are served by user documents:

- ❑ Assists user to complete the task
- ❑ Facilitates the user training
- ❑ Helps user to explore the advanced system capabilities
- ❑ Helps in troubleshooting problems
- ❑ Helps in undoing the user's mistakes
- ❑ User can understand the structure of the system

Training

- User – to know how to operate the new software
- Various training sessions are arranged

  Low- level /Middle level/ Top level of the management

- Reasons for need of training
  - User –will handle new system efficiently
  - Increases productivity
  - Implementation of system can be done fast
  - Less chances of system failure.

Conversion *[Changeover or cutover]*

- Process of changing the old system to a new or modified system
- 4 method of Conversion Process

Parallel Conversion

- Processing of data is carried out in both(new/old system) for a specific trial period.
- No loss in the revenue of business

Direct Conversion

- User can Switch directly from Old System to New System
- New System Fully Replaces the old system/User do no use old System

Pilot Conversion

- Isolated unit in the organization is selected for testing the new system
- The user have knowledge are piloting a new system and suggest changes.

Phased Conversion

- If it is difficult to install the new system, this method is used
- Implementation is staged over a period of time
- May extend few weeks to even months

5. Maintenance

- The maintenance function begins as soon as the system has been implemented in the business
- It involves – Evaluating, monitoring and modifying the operations of the system.
- When new system is implemented – "Learning Curve" takes place.
- Objectives of System Maintenance:
  - To Correct Errors
  - To Keep System Current (Update)
  - To Improve the System

Learning Curve:
People who are working with the system for the first time make lot of mistakes, as they are unfamiliar with the features of the system.

Categories of Maintenance:

Corrective Maintenance:
When failure is associated with the functioning of the system

Adaptive Maintenance:
The function of the program is changed – to work in desired manner

Preventive Maintenance:
The function of the program is changed – to work in desired manner

Perfective Maintenance:
Adding new programs/modifying existing
User's additional needs
Changes within or outside organization

Predictive Maintenance:
Strategic change made in anticipation of likely changes to technology

Advantages of SDLC

- Easy to be explained
- Stages and activities are well defined
- Redundancies and inefficiencies can be reduced
- Systematic approach, formally reviewed at the end of every phase
- Error detection: As SDLC is well defined, it is possible to cover the errors and output is maximized
- Process and results are well documented

Disadvantages of SDLC

- **Inflexible**: If system requirements change rapidly, SDLA –unable to adapt
- **Time Consuming** : To implement changes, the whole process has to be restarted
- **Lack of Changes**: The model prevents changes, instead of accommodating them
- **Costly** and required more time, in addition to detailed plan
- **No early prototype** of the software are produced
- **Implementation** of User Needs

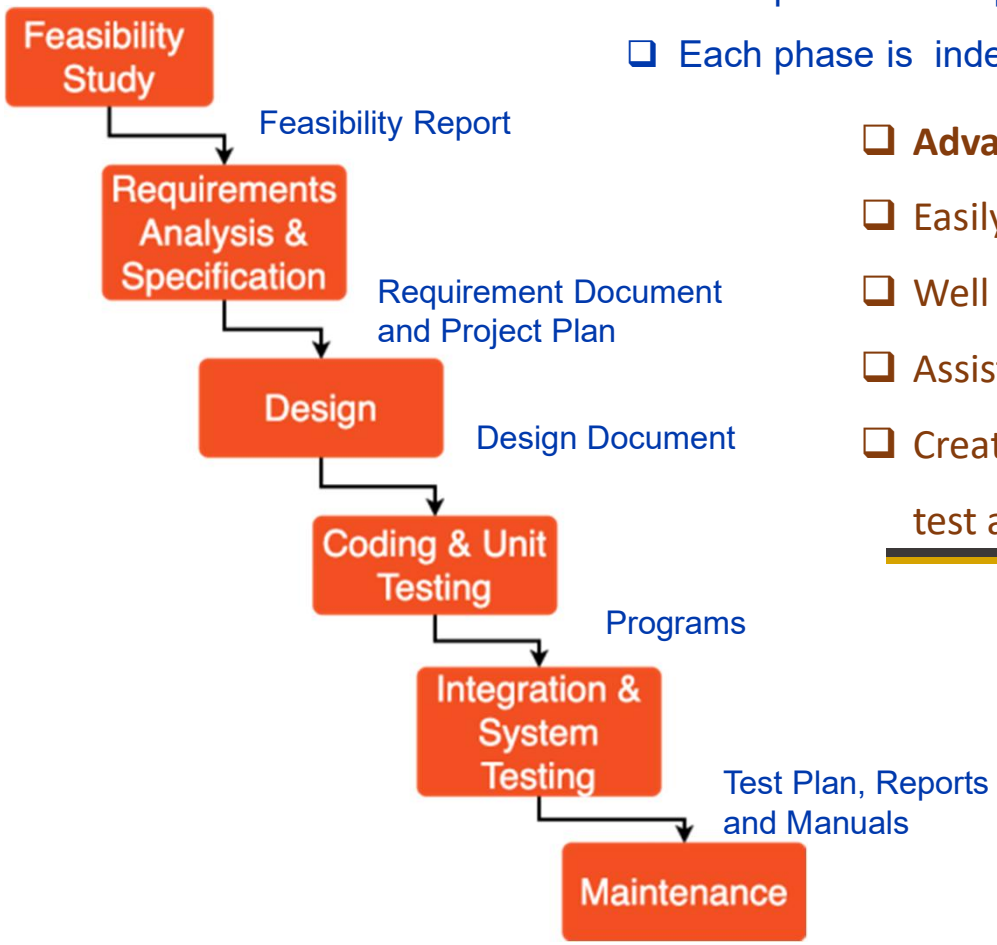
Waterfall Model

- The classical Waterfall model is not a practical model
- It cannot be used to develop the software projects in the real world.
- It is a theoretical way of developing a software model.
- All the lifecycle models are derived from the classical waterfall model
- It is also referred as the " **Classical Life Cycle Model**", " **Linear, sequential model**" or simply waterfall model.
- It involves a systematic, sequential approach to software development at system level and moves through analysis, design, coding, testing and support.

Waterfall Model Cont'd …

Phases of the Classical Waterfall Model

- Each phase is completed first and reviewed before moving onto the next phase.
- Each phase is independent of the other

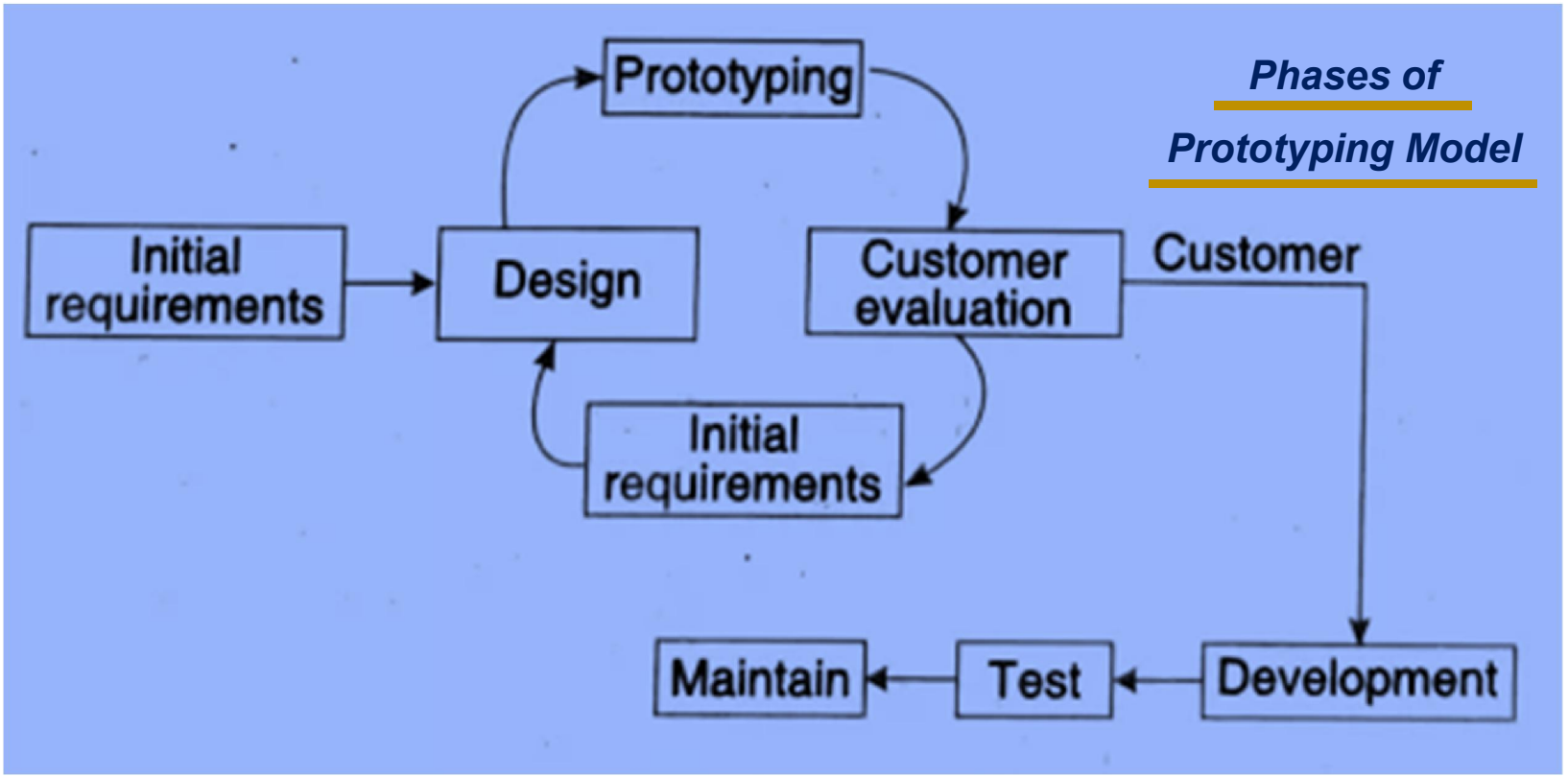


- **Advantages of Waterfall Model:**
- Easily understandable
- Well defined inputs and outputs for each phase
- Assists project manager in different project planning
- Creates a model into which different methods of analysis, design, code, test and support is tested

- **Disadvantages of Waterfall Model:**
- Sequential in nature –move forward – not backward
- No overlapping and interaction between phases
- Project team has very little interaction with users
- Delivery of the system is not done in pieces
- Have vague specifications, not suitable for new projects

Prototyping Model

- Early approximation of the final software product
- Prototypes of the software are created, tested and rectified until final accepted prototype is achieved.



Phases of Prototyping Model

Three types

1. **Exploratory Development**
2. **Evolutionary Prototyping**
3. **Throwaway Prototyping**

Prototyping Model

Exploratory Development

Software development model – mainly involves in planning and creating **different designs** until close to acceptable product is achieved

Evolutionary Prototyping

Split the system into several independent modules.

Developing actual product with minimal functionalities meeting basic requirements

Further features are added afterwards in future

Throwaway Prototyping

- Also known as **Rapid** or **Close ended** prototyping model
- **Small part** of software is developed and given to end user to **evaluate**
- Based on **feedback**, final product is developed and **prototype is discarded**

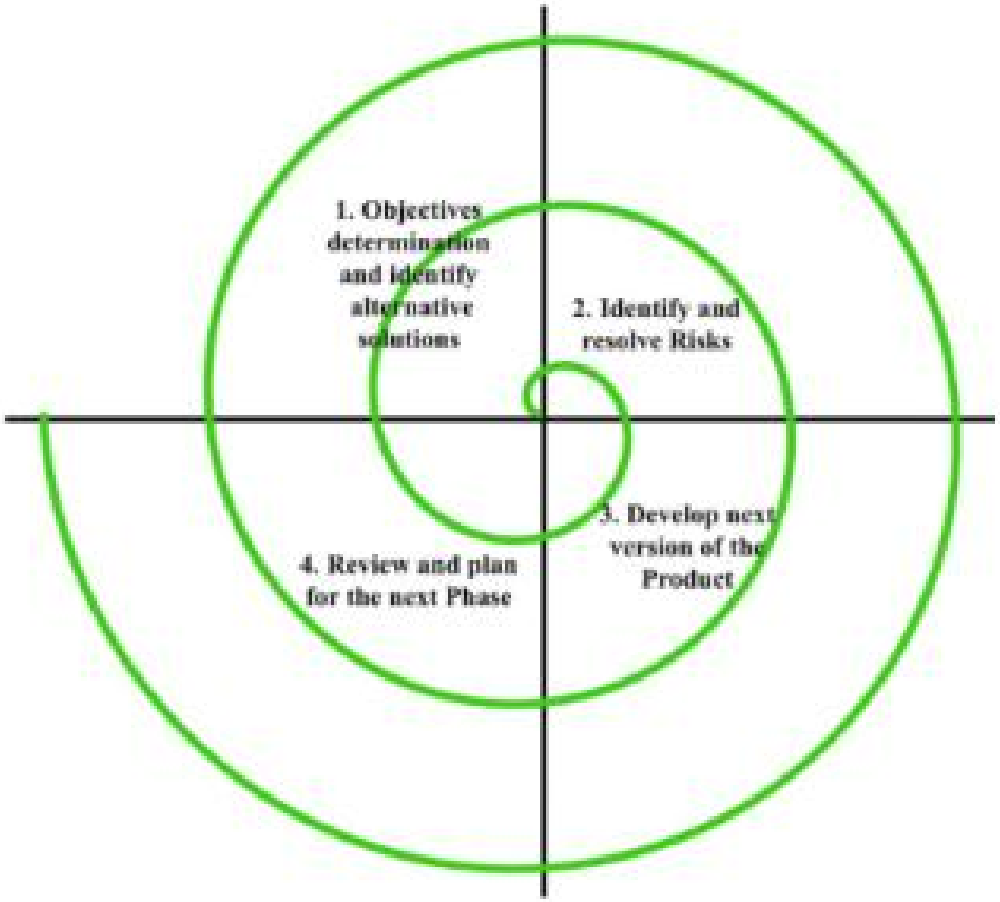
Advantages of Prototyping model:

- **Few features only- development time is less**
- **Precise user requirements-available for final product development**
- **High user acceptance (S/W product already evaluated by user**

Disadvantages of Prototyping model:

- If initial prototype is not satisfactory for user – **loss of interest** for the product
- Customers sometimes demand the actual product to be **delivered soon** after seeing an early prototype.
- This model is **costly**.

Spiral Model



- Boehm proposed the spiral Life cycle model
- Combination of Waterfall model+Risk management techniques
- Each round of spiral
  - Recognizes the highest risk sub-problem
  - Provides solution to the problem
- In its diagrammatic representation, it looks like a spiral with many loops
- Each loop of the spiral is called a Phase of the software development process.
- This model has capability to handle risks.
- is also known as Meta Model because it subsumes all the other SDLC models.
- Number of phases is not fixed.

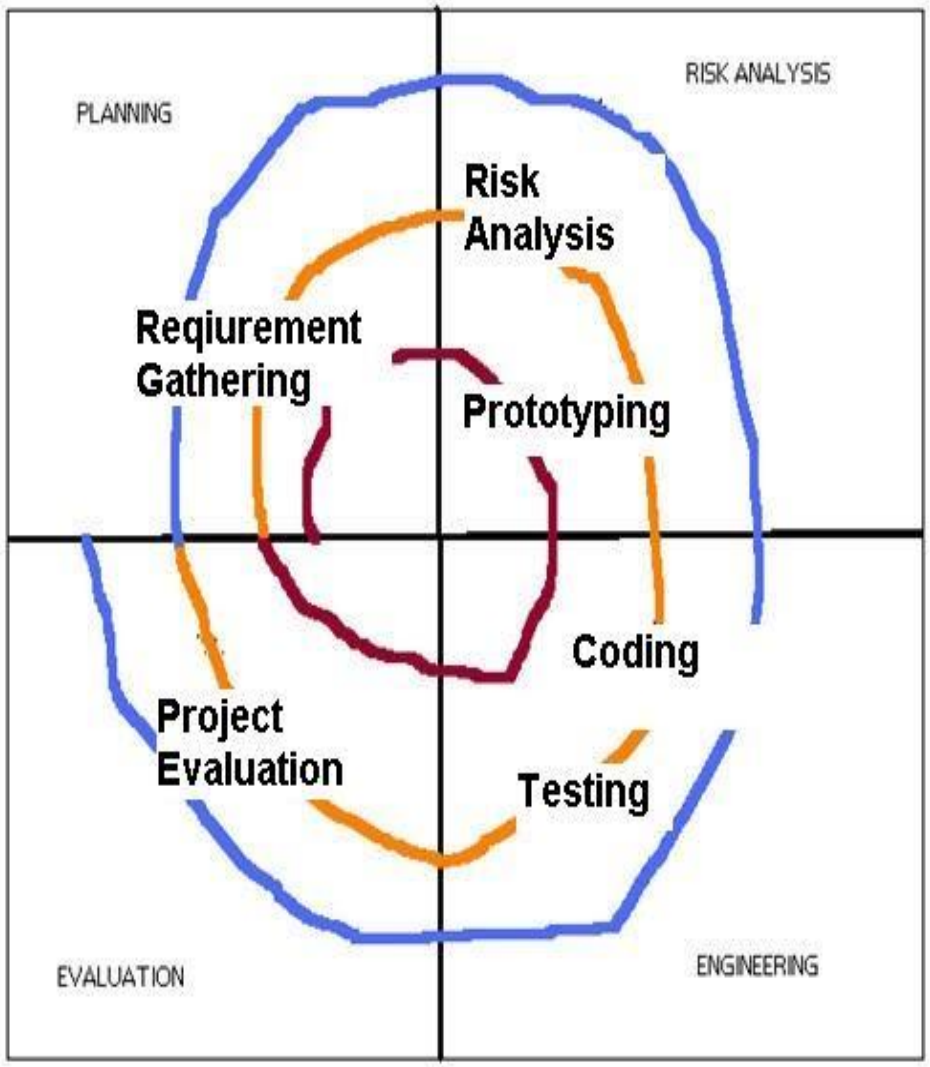
Quadrants in Spiral Model

First Quadrant

- To identify the objective of the phase and different solutions are considered

Fourth Quadrant

- Reviews the results of previous stage with customer
- Iteration is done at every stage of the spiral to come up with complete s/w version



Second Quadrant

- Evaluation of various alternatives.
- Focusses on risks involved
- This will hinder the completion of project

Third Quadrant

- Developing strategies to tackle risks
- Strategies includes benchmarking, simulation and prototyping

Spiral Model

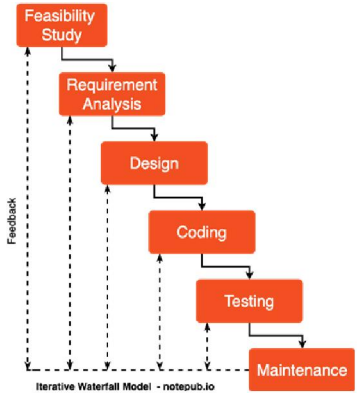
Advantages

- Risk identification at early stage at low costs
- Visible prototype to user- can view prototype at initial stage
- Suitable for high risk projects
- Flexile for adding functionality

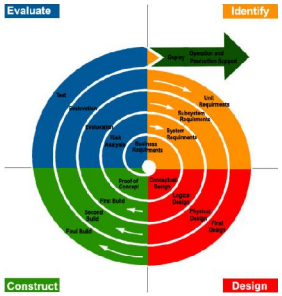
Disadvantages

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

Difference between Spiral and waterfall Model

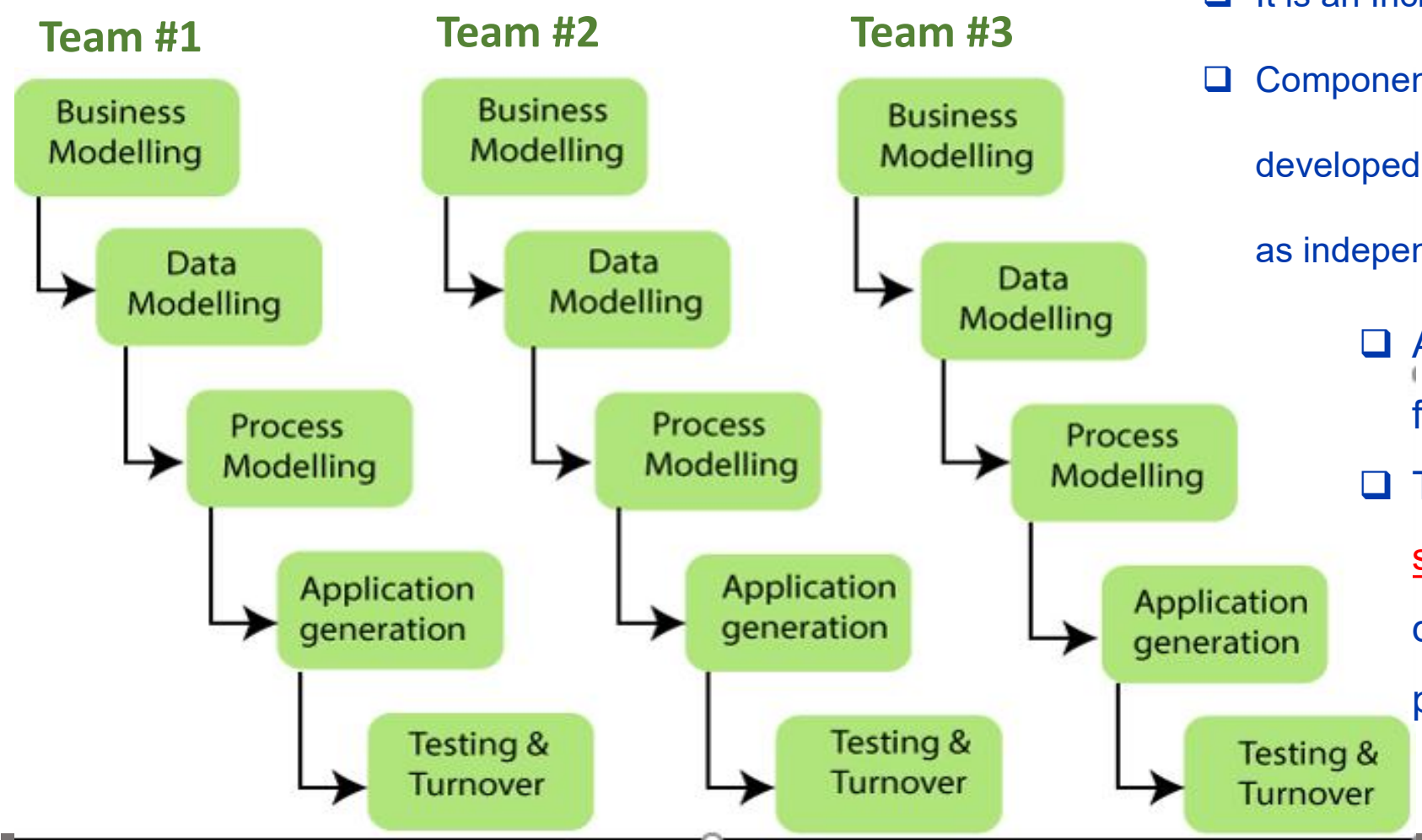


Spiral Model	Waterfall Model
Not suitable for small projects	Suitable for small projects
Better risk management	High amount of risk and uncertainty
Complex process	Easy to understand
Process may go indefinitely	Stages are clearly defined
Suitable for long & ongoing projects	Not suitable for long & ongoing projects
Iterations are followed	Sequence is followed
Flexible with user requirements	Requirements once fixed- can not be modified
Refinements are easily possible	Refinements are not easy
Phases are repeated itself	Phases are processed and completed one at a time



RAD Model

Rapid Application Development Model



- It is an Incremental Model
- Components or functionalities are developed in parallel considering them as independent small- sub-project
  - Afterwards integrated into final product.
- To develop software in small span of time by delivering small sub-project.

Business Modeling:

In this phase, business functions and product scope are decided during various meetings between the requirements planning team and the client team.

Data Modeling: In the data modeling phase, all the information derived in the business modeling phase is analyzed and separated into different data elements important for the business.

Process Modeling – In this phase, all the data objects gathered in the process modeling phase are transformed into useful information.

Application Generation – In this stage, the actual prototype is developed using different automated CASE tools.

Testing and Turnover – In this stage, all the modules and interfaces of the prototype are tested.