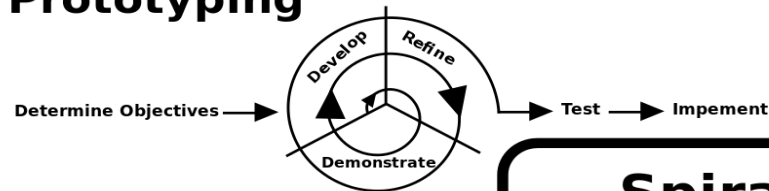
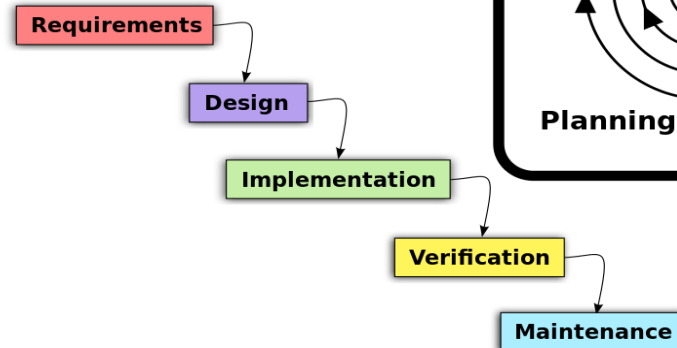


# SOFTWARE PROCESS

## Prototyping



## Waterfall



## Spiral



# SOFTWARE PROCESS

## **Process**

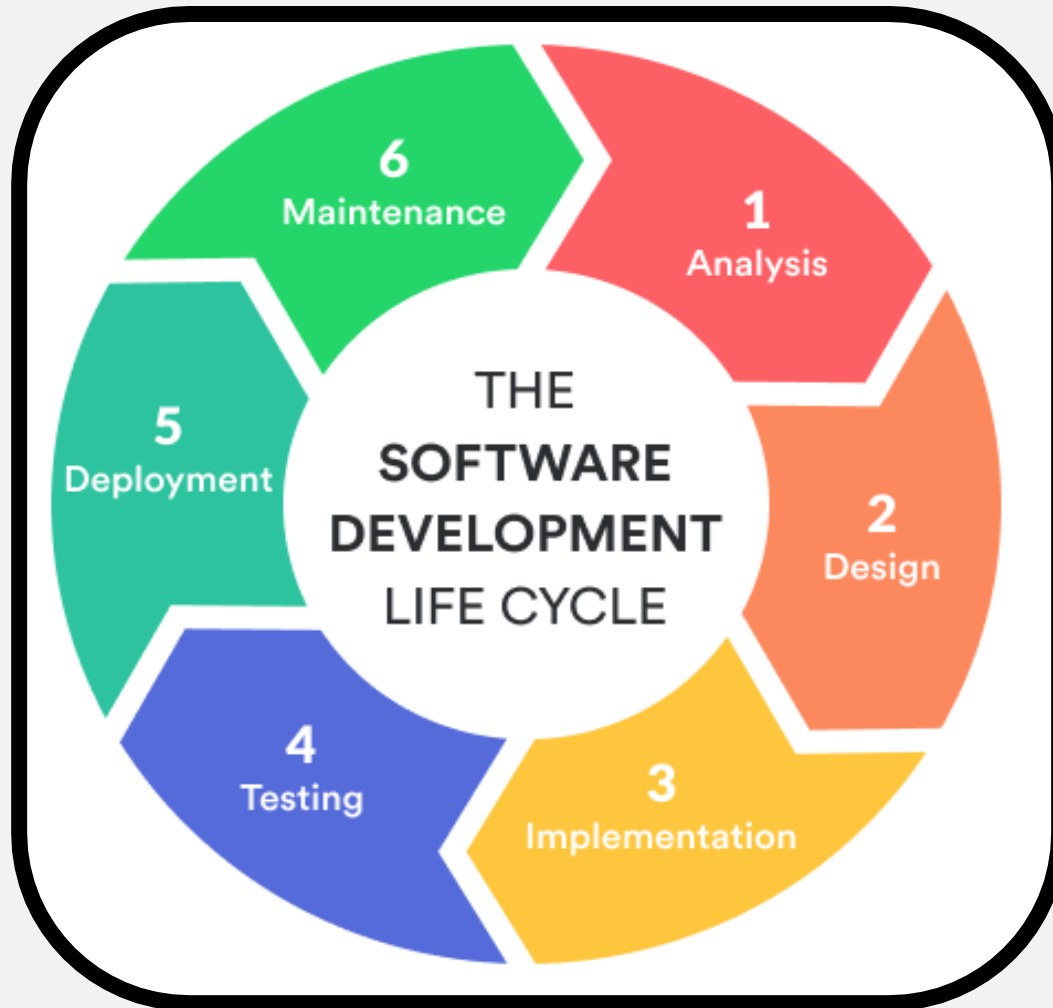


Series of predictable steps-a road map that helps create a timely and high quality entity



Software Process is a framework for the tasks that are required to build high quality software

# SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)



# **SDLC: STRUCTURED ANALYSIS**

## **1. Data Modeling**

- a. Entity Relation Diagram (ERD)
- b. Data Dictionary (DD)

## **2. Functional Modeling**

- 1. Data Flow Diagram (DFD)
- 2. Process Specification (PS) and Mini Specification (MS)

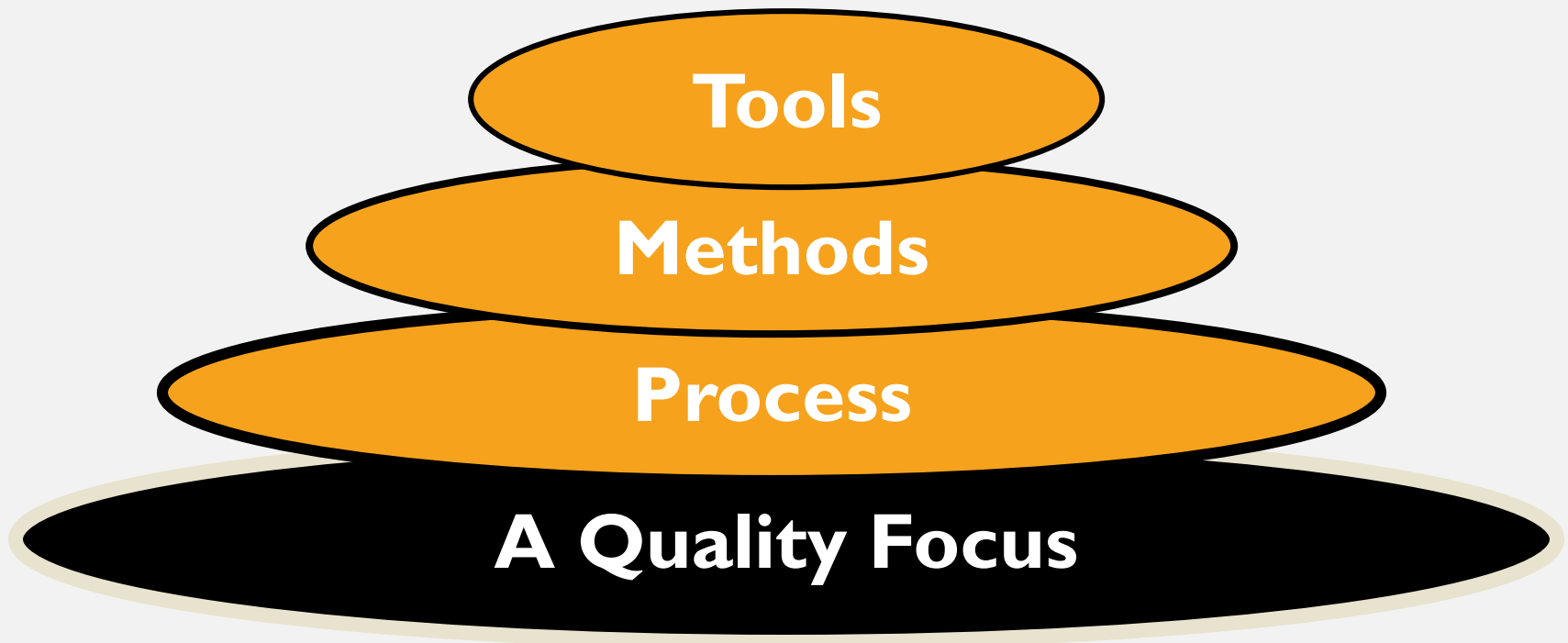
## **3. Behavioral Modeling**

- 1. State Transition Diagram
- 2. Control Specification (CS)

# **SDLC: STRUCTURED DESIGN**

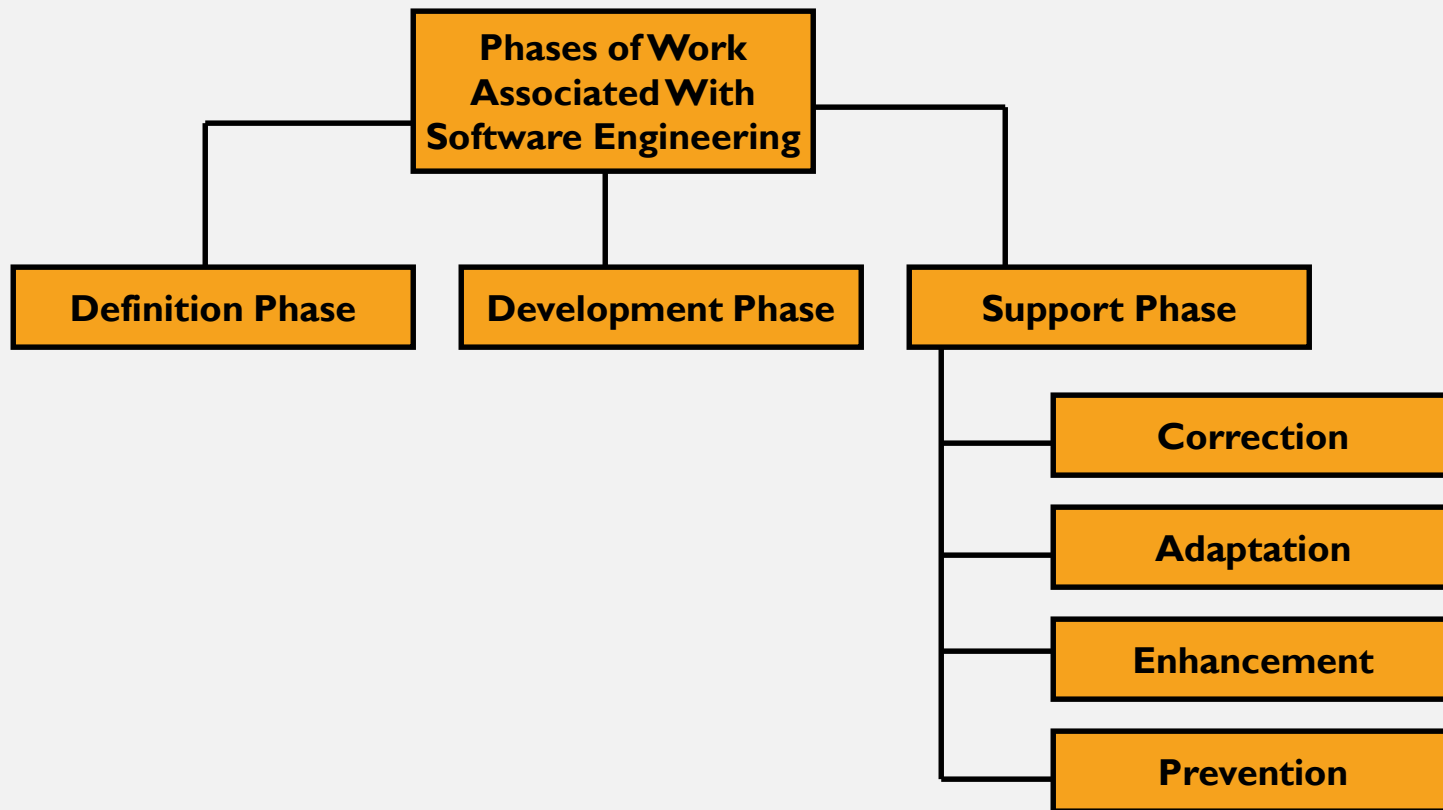
- 1. Data Design**
- 2. Architectural Design**
- 3. Component Level Design**
- 4. User Interface Design**

# **SOFTWARE ENGINEERING:** **A LAYERED TECHNOLOGY**

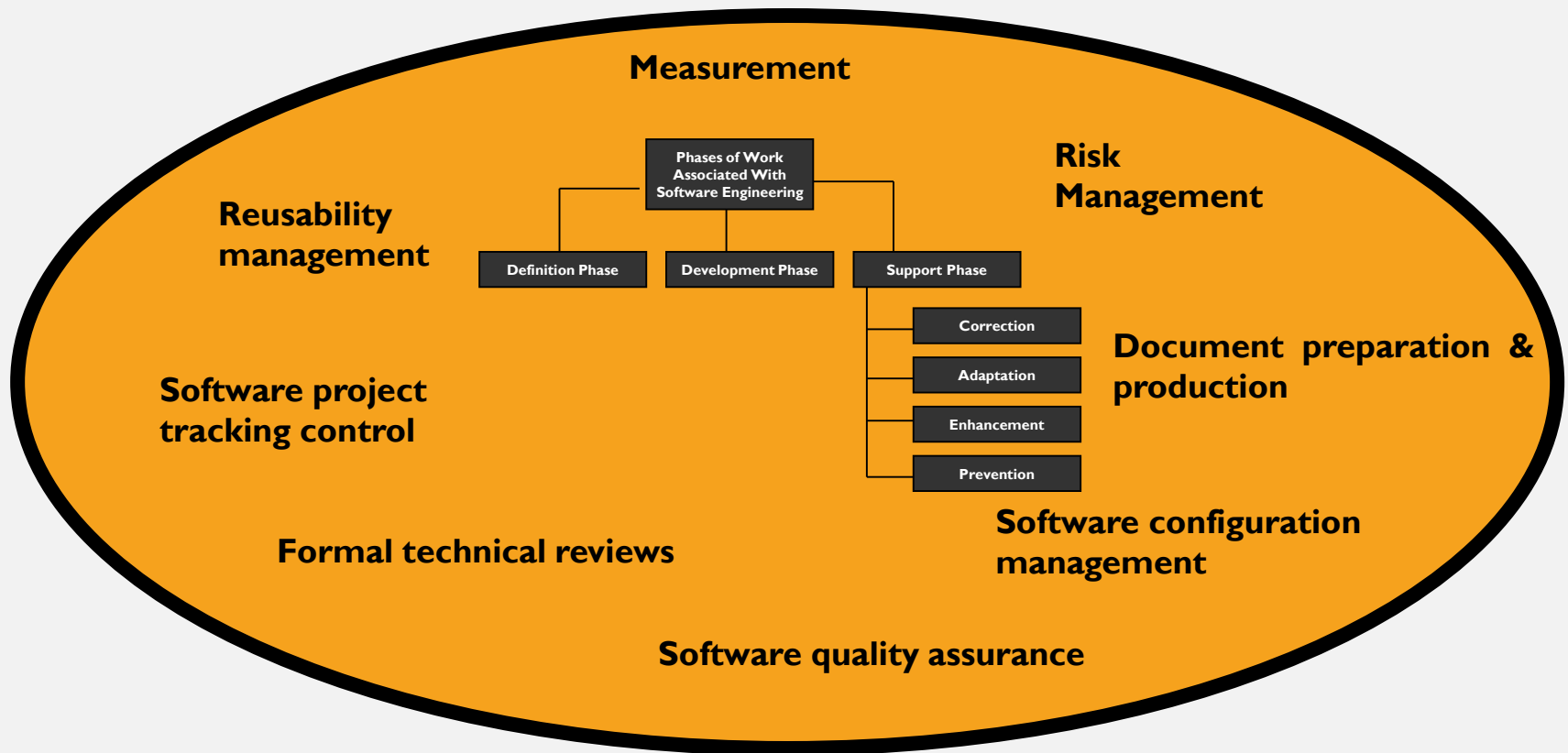


# **SOFTWARE ENGINEERING**

## **A GENERIC VIEW**



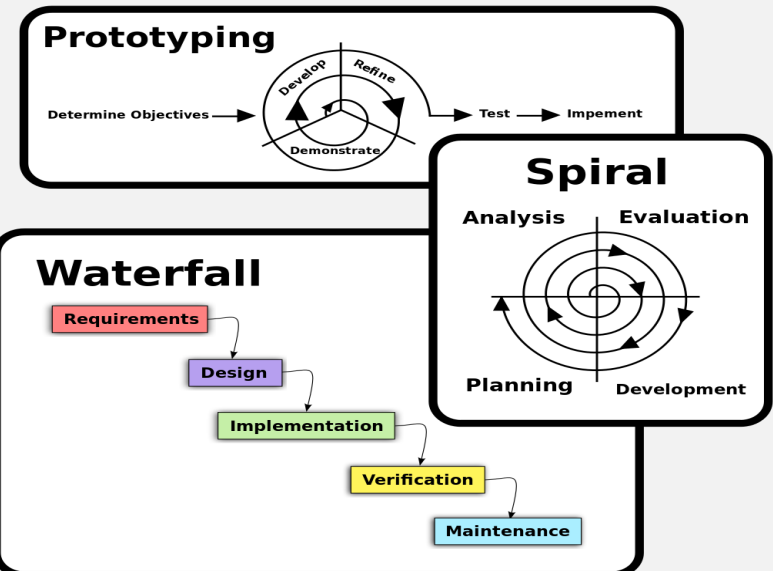
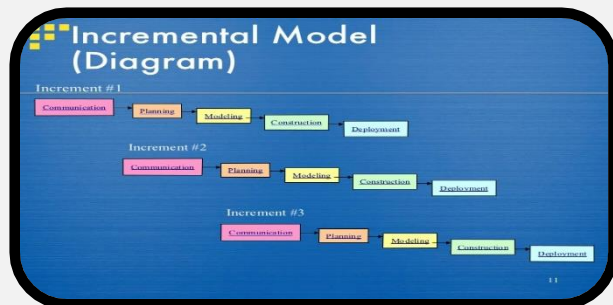
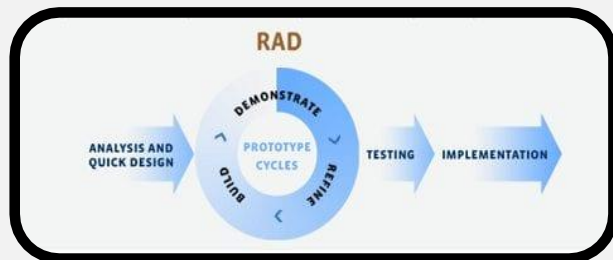
# Software Engineering: A Generic View





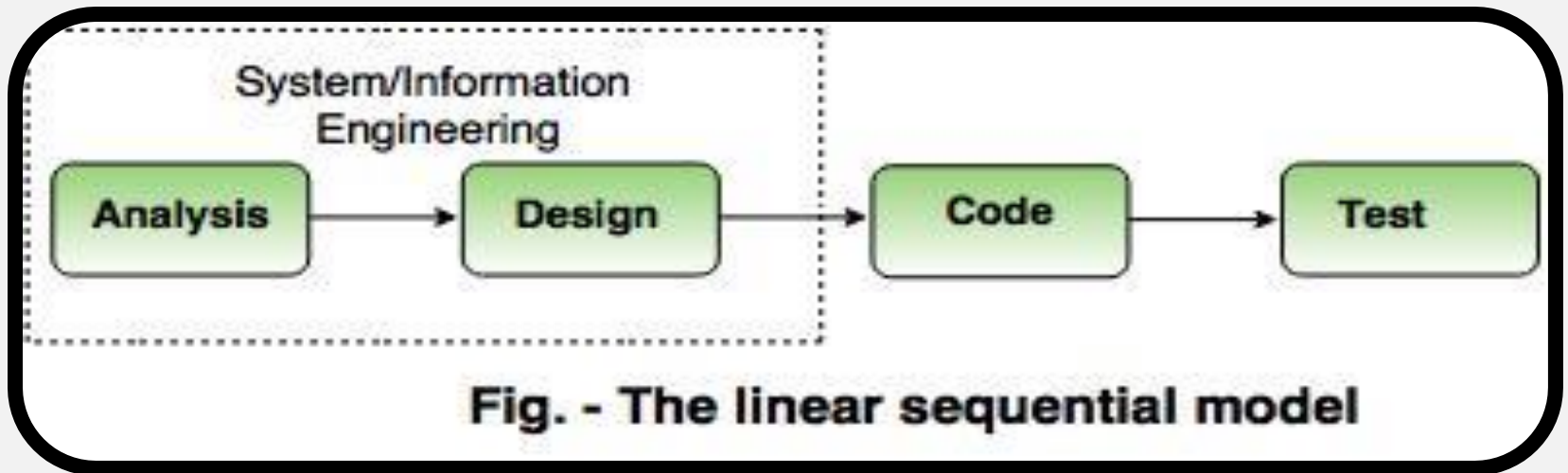
# SOFTWARE PROCESS MODELS

1. Linear Sequential Model (LSM)
2. Prototyping Model
3. Rapid Application Development Model (RAD)
4. Incremental Model
5. Spiral Model

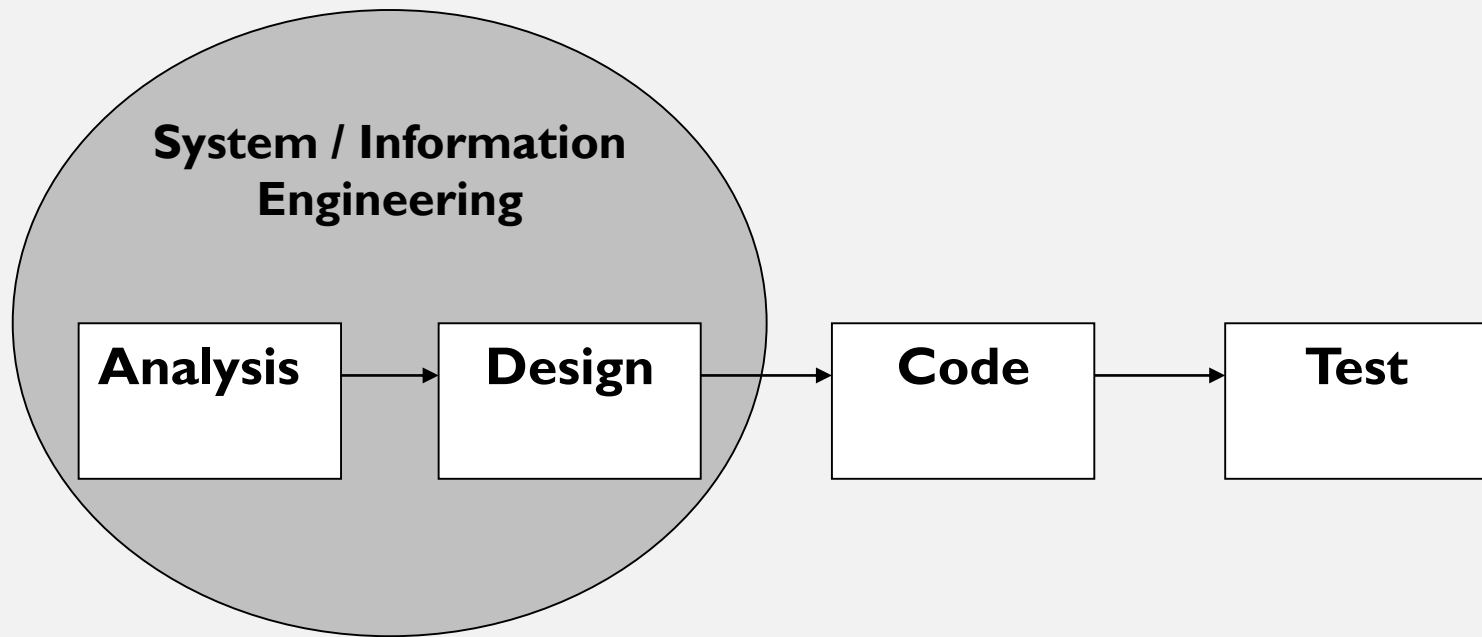


# Linear Sequential model

Also known as the classic life cycle or waterfall model, it suggests a systematic, sequential approach to software development that begins at the system level and progress through analysis, design, coding, testing and support.



# LINEAR SEQUENTIAL MODEL



# **LINEAR SEQUENTIAL MODEL**

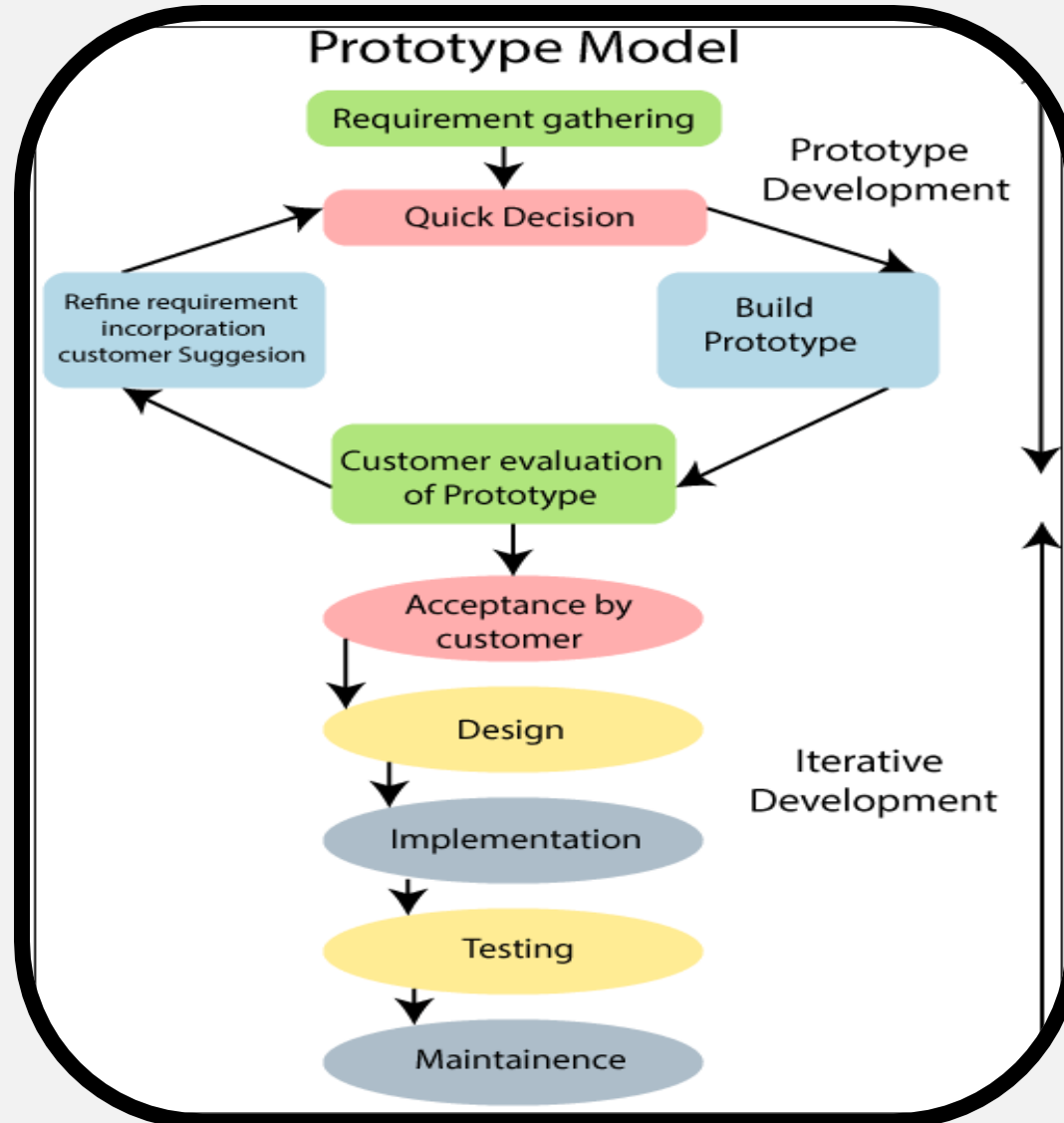
## **Limitations**

- Real projects rarely follow the sequential flow and changes can cause confusion.
- This model has difficulty accommodating requirements change.
- The customer will not see a working version until the project is nearly complete.
- Developers are often blocked unnecessarily, due to previous tasks not being done.

## **PROTOTYPING MODEL**

The developer and customer define the overall objectives for the software. A quick design focuses on what the customer will see. From this, a prototype is constructed. The user evaluates it and improvements are made. This continues in an iterative fashion until a satisfactory product is achieved.

# Prototyping:



# **PROBLEMS WITH PROTOTYPING**

- The customer sees a working version and expects the finished product to be available in a short time. This puts pressure on the developer to take short cuts, at the expense of quality and maintainability.
- The developer may make compromises for speed.
- Inappropriate tools may be used or inefficient algorithms may be used, which then become integral parts of the system.
- If the user isn't focused on what they want, the system may never be completed.

# **RAPID APPLICATION DEVELOPMENT MODEL (RAD)**

- Rapid Application Development is a linear sequential software development process model that emphasises an extremely short development cycle.
- A component-based construction approach is used.
- To use this approach, the project scope must be constrained and the requirements should be well understood.
- A task that should take no more than ninety days to complete is modelled, generated and implemented.
- There can be several teams working on different components during this ninety day time-box.



# THE RAD MODEL

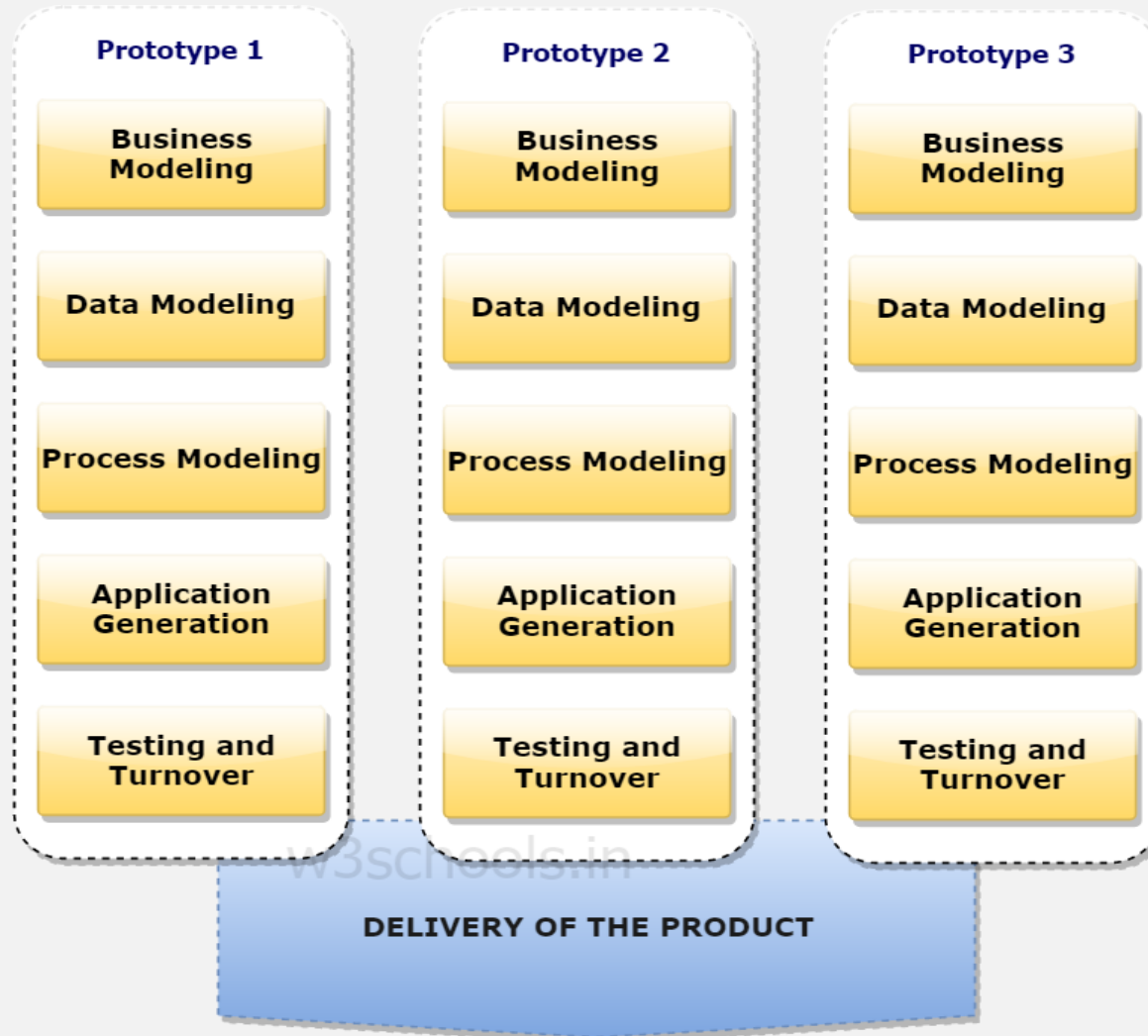


Fig: SDLC RAD Model

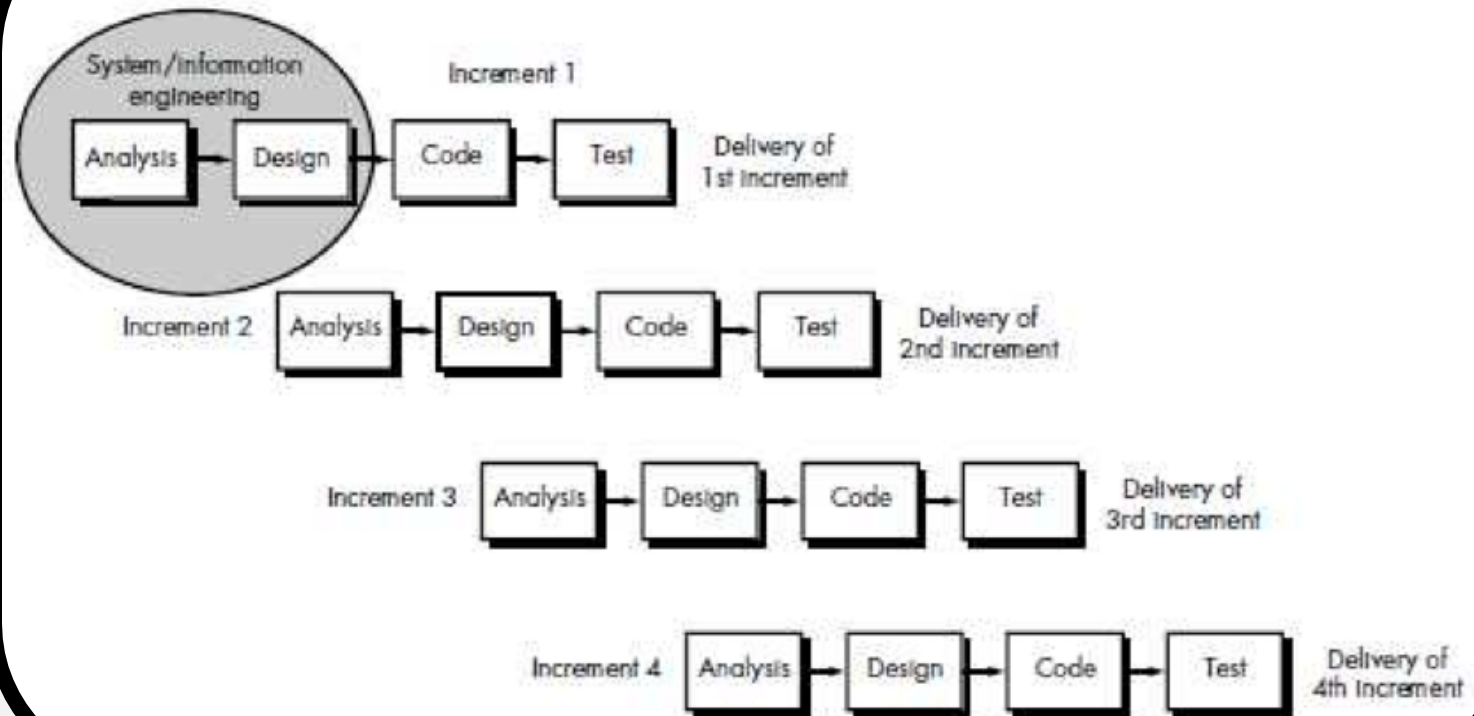
# **PROBLEMS WITH RAD**

- For large, scalable projects, RAD requires sufficient human resources to create the right number of RAD teams
- RAD requires developers and customers who are committed to the rapid-fire activities necessary to complete a system in this time frame, or failure will result.

# **THE INCREMENTAL MODEL**

- This is a combination of the linear sequential model and the iterative model.
- The problem is broken into increments, and each increment is tackled as a linear sequence.
- Further increments can either be done after the previous ones, or can overlap with the previous ones.
- Incremental delivery focuses on the delivery of an operational product with each increment.
- Early increments are stripped-down versions of the final product.

# INCREMENTAL MODEL



# EVOLUTION OF MS WINDOWS

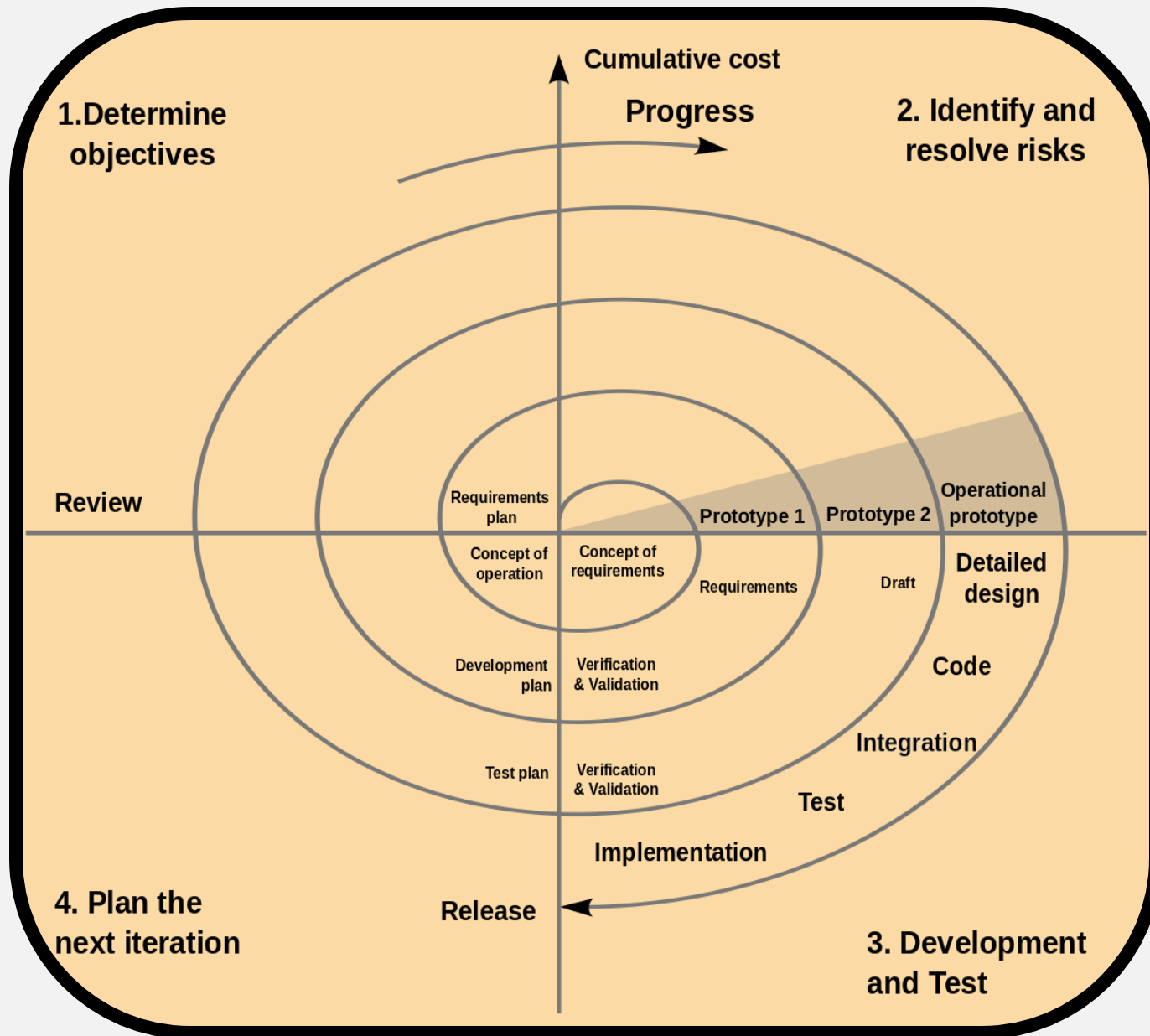
Sr. #	MS Windows Version	Year	Features: Added or changed
1	Windows 1	1985	
2	Windows 2	1987	
3	Windows 3	1990	
4	Windows 3.1	1992	
5	Windows 95	1995	
6	Windows 98	1998	
7	Windows Me	2000	
8	Windows 2000	2000	
9	Windows XP	2001	
10	Windows Vista	2007	
11	Windows 7	2009	
12	Windows 8	2012	
13	Windows 8.1	2013	
14	Windows 10	2014	

## **ADVANTAGES OF INCREMENTAL MODEL**

- Less staffing is required than in a RAD project
- Early delivery is guaranteed
- Progress of the whole project is not delayed if one of the resources is not available for part of it

# **THE SPIRAL MODEL:**

- Boehm's (1988) spiral model couples the iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model.
- Software is developed in a series of incremental releases.
- During the early releases, there may be just a paper model, but the system becomes increasingly more complete.
- There are a number of framework activities (Customer communication, Planning, Risk analysis, Engineering, Construction and release, Customer evaluation).
- Unlike any of the other models, this model keeps revisiting the system throughout its lifetime.





# **CONCLUDING REMARKS:**

- There are a variety of process models, each of which can be used successfully.
- Once a process model has been used to develop a system, documentation style, organisation and structure should either remain in the format of that process model, or all be converted to a different process model.
- This is particularly important where automated tools are used.

**THANK**  
**YOU!**