

## **UNIT - I**

### **INTRODUCTION:**

Software Engineering is a framework for building software and is an engineering approach to software development. Software programs can be developed without S/E principles and methodologies but they are indispensable if we want to achieve good quality software in a cost effective manner.

Software is defined as:

Instructions + Data Structures + Documents

Engineering is the branch of science and technology concerned with the design, building, and use of engines, machines, and structures. It is the application of science, tools and methods to find cost effective solution to simple and complex problems.

**SOFTWARE ENGINEERING** is defined as a systematic, disciplined and quantifiable approach for the development, operation and maintenance of software.

### **The Evolving role of software:**

The dual role of Software is as follows:

1. A Product- Information transformer producing, managing and displaying information.
2. A Vehicle for delivering a product- Control of computer(operating system),the communication of information(networks) and the creation of other programs.

### **Characteristics of software**

- **Software is developed or engineered**, but it is not manufactured in the classical sense.
- **Software does not wear out**, but it deteriorates due to change.
- **Software is custom built** rather than assembling existing components.

### **THE CHANGING NATURE OF SOFTWARE**

The various categories of software are

1. System software
2. Application software
3. Engineering and scientific software
4. Embedded software
5. Product-line software
6. Web-applications
7. Artificial intelligence software

- **System software.** System software is a collection of programs written to service other programs
- **Embedded software**-- resides in read-only memory and is used to control products and systems for the consumer and industrial markets.
- **Artificial intelligence software.** Artificial intelligence (AI) software makes use of nonnumeric algorithms to solve complex problems that are not amenable to computation or straightforward analysis

**Engineering and scientific software.** Engineering and scientific software have been characterized by "number crunching" algorithms.

### **LEGACY SOFTWARE**

Legacy software are older programs that are developed decades ago. The quality of legacy software is

poor because it has inextensible design, convoluted code, poor and nonexistent documentation, test cases and results that are not achieved.

As time passes legacy systems evolve due to following reasons:

- The software must be adapted to meet the needs of new computing environment or technology.
- The software must be enhanced to implement new business requirements.
- The software must be extended to make it interoperable with more modern systems or database
- The software must be rearchitected to make it viable within a network environment.

## **SOFTWARE APPLICATIONS**

### **System Software –**

System Software is necessary to manage the computer resources and support the execution of application programs. Software like operating systems, compilers, editors and drivers, etc., come under this category. A computer cannot function without the presence of these. Operating systems are needed to link the machine-dependent needs of a program with the capabilities of the machine on which it runs. Compilers translate programs from high-level language to machine language.

### **Application Software –**

Application software is designed to fulfill the user's requirement by interacting with the user directly. It could be classified into two major categories:- generic or customized. Generic Software is the software that is open to all and behaves the same for all of its users. Its function is limited and not customized as per the changing requirements of the user. However, on the other hand, Customized software the software products which are designed as per the client's requirement, and are not available for all.

### **Networking and Web Applications Software –**

Networking Software provides the required support necessary for computers to interact with each other and with data storage facilities. The networking software is also used when software is running on a network of computers (such as the World Wide Web). It includes all network management software, server software, security and encryption software, and software to develop web-based applications like HTML, PHP, XML, etc.

### **Embedded Software –**

This type of software is embedded into the hardware normally in the Read-Only Memory (ROM) as a part of a large system and is used to support certain functionality under the control conditions.

Examples are software used in instrumentation and control applications like washing machines, satellites, microwaves, etc.

### **Reservation Software –**

A Reservation system is primarily used to store and retrieve information and perform transactions related to air travel, car rental, hotels, or other activities. They also provide access to bus and railway reservations, although these are not always integrated with the main system. These are also used to relay computerized information for users in the hotel industry, making a reservation and ensuring that the hotel is not overbooked.

### **Business Software –**

This category of software is used to support business applications and is the most widely used category of software. Examples are software for inventory management, accounts, banking, hospitals, schools, stock markets, etc.

### **Entertainment Software –**

Education and entertainment software provides a powerful tool for educational agencies, especially those that deal with educating young children. There is a wide range of entertainment software such as computer games, educational games, translation software, mapping software, etc.

### **Artificial Intelligence Software –**

Software like expert systems, decision support systems, pattern recognition software, artificial neural networks, etc. come under this category. They involve complex problems which are not affected by complex computations using non-numerical algorithms.

### **Scientific Software –**

Scientific and engineering software satisfies the needs of a scientific or engineering user to perform enterprise-specific tasks. Such software is written for specific applications using principles, techniques, and formulae specific to that field. Examples are software like MATLAB, AUTOCAD, PSPICE, ORCAD, etc.

### **Utilities Software –**

The programs coming under this category perform specific tasks and are different from other software in terms of size, cost, and complexity. Examples are anti-virus software, voice recognition software, compression programs, etc.

### **Document Management Software –**

Document Management Software is used to track, manage and store documents in order to reduce the paperwork. Such systems are capable of keeping a record of the various versions created and modified by different users (history tracking). They commonly provide storage, versioning, metadata, security, as well as indexing and retrieval capabilities.

## **A GENERIC VIEW OF PROCESS SOFTWARE ENGINEERING-A LAYERED TECHNOLOGY**



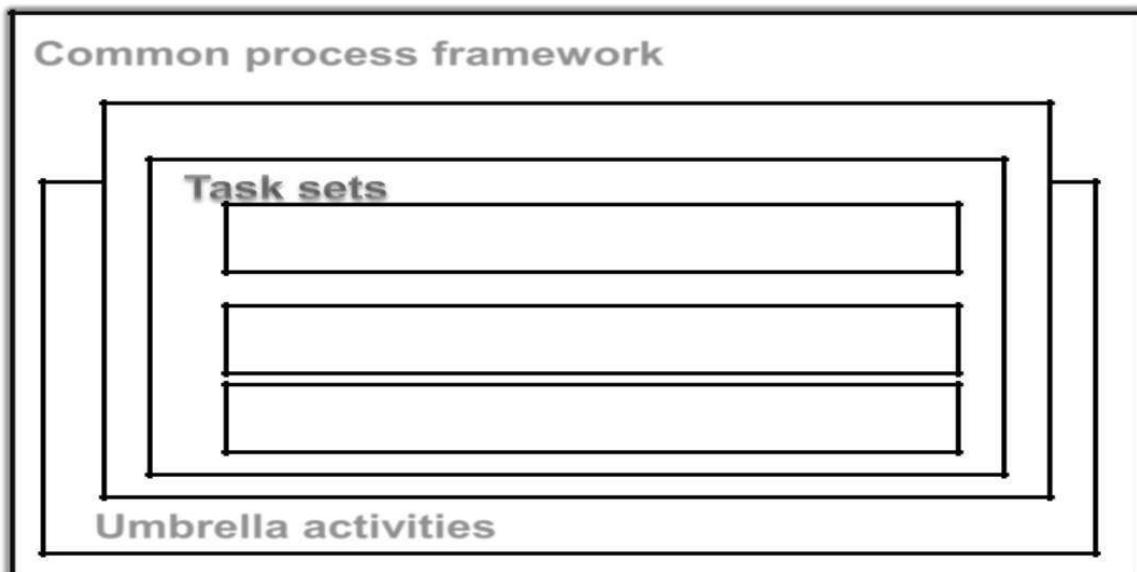
Fig: Software Engineering-A layered technology

## **SOFTWARE ENGINEERING - A LAYERED TECHNOLOGY**

- Quality focus - Bedrock that supports Software Engineering.
- Process - Foundation for software Engineering
- Methods - Provide technical How-to's for building software
- Tools - Provide semi-automatic and automatic support to methods

## A PROCESS FRAMEWORK

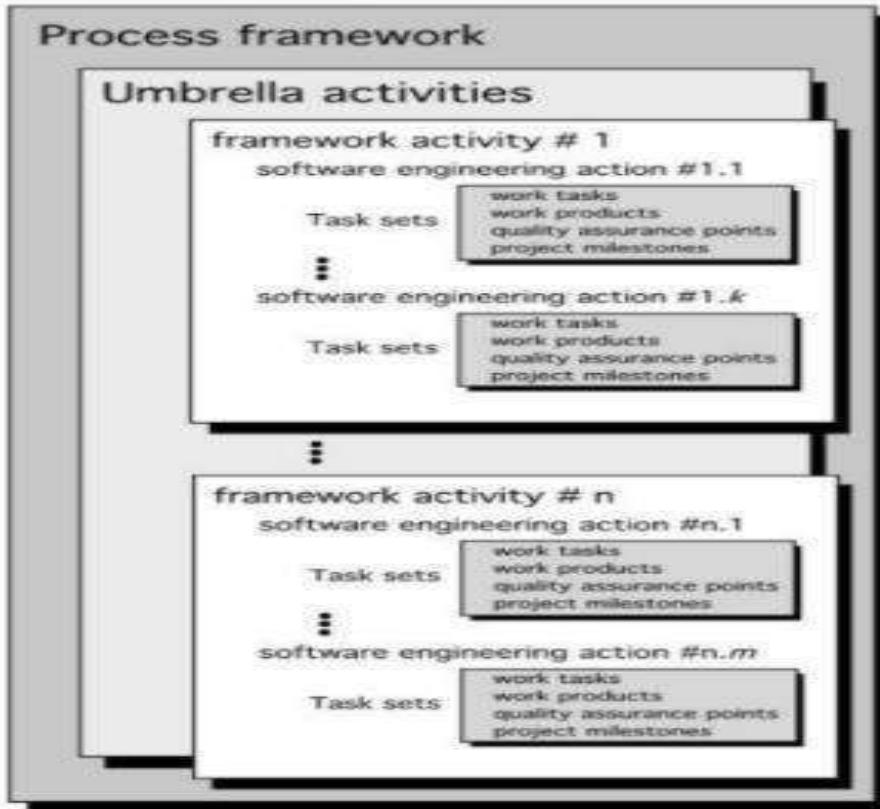
- Establishes the foundation for a complete software process
- Identifies a number of framework activities applicable to all software projects
- Also include a set of umbrella activities that are applicable across the entire software process.



A PROCESS FRAMEWORK comprises of :

Common process framework Umbrella activities Framework activities Tasks, Milestones, deliverables SQA points

## Software process



## A PROCESS FRAMEWORK

Used as a basis for the description of process models Generic process activities

- Communication
- Planning
- Modeling
- Construction
- Deployment

## A PROCESS FRAMEWORK

Generic view of engineering complimented by a number of umbrella activities

- Software project tracking and control
- Formal technical reviews
- Software quality assurance
- Software configuration management
- Document preparation and production
- Reusability management
- Measurement
- Risk management

## CAPABILITY MATURITY MODEL INTEGRATION(CMMI)

- Developed by SEI(Software Engineering institute)
- Assess the process model followed by an organization and rate the organization with different levels
- A set of software engineering capabilities should be present as organizations reach different levels of process capability and maturity.

CMMI process meta model can be represented in different ways

- 1.A continuous model
- 2.A staged model

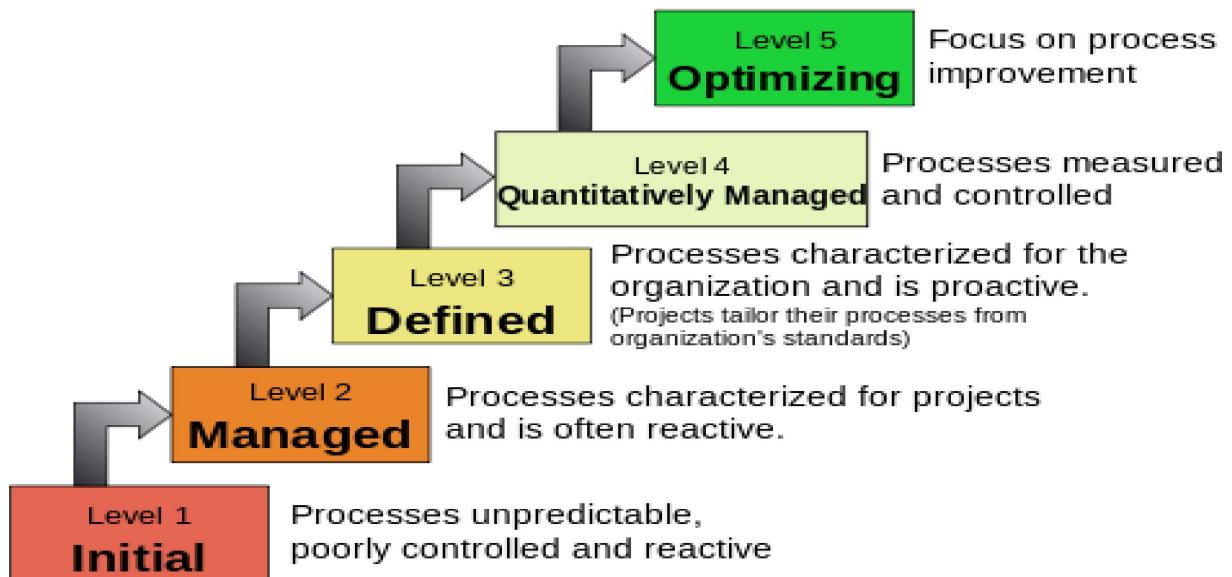
#### Continuous model:

- Lets organization select specific improvement that best meet its business objectives and minimize risk-Levels are called capability levels.
- Describes a process in 2 dimensions
- Each process area is assessed against specific goals and practices and is rated according to the following capability levels.

#### **Six levels of CMMI**

- Level 0:Incomplete
- Level 1:Performed
- Level 2:Managed
- Level 3:Defined
- Level 4:Quantitatively managed
- Level 5:Optimized

## **Characteristics of the Maturity levels**



- Incomplete -Process is adhoc . Objective and goal of process areas are not known
- Performed -Goal, objective, work tasks, work products and other activities of software process are carried out
- Managed -Activities are monitored, reviewed, evaluated and controlled
- Defined -Activities are standardized, integrated and documented
- Quantitatively Managed -Metrics and indicators are available to measure the process and quality
- Optimized - Continuous process improvement based on quantitative feed back from the user
- Use of innovative ideas and techniques, statistical quality control and other methods for process improvement

### CMMI - Staged model

- This model is used if you have no clue of how to improve the process for quality software.
- It gives a suggestion of what things other organizations have found helpful to work first
- Levels are called maturity levels

### PROCESS PATTERNS

Software Process is defined as collection of Patterns. Process pattern provides a template. It comprises of

- Process Template
- Pattern Name
- Intent
- Types
  - Task pattern
  - Stage pattern
  - Phase Pattern
- Initial Context
- Problem
- Solution
- Resulting Context
- Related Patterns

### PROCESS ASSESSMENT

Does not specify the quality of the software or whether the software will be delivered on time or will it stand up to the user requirements. It attempts to keep a check on the current state of the software process with the intention of improving it.

PROCESS ASSESSMENT Software Process Software Process Assessment Software Process Improvement Motivates Capability determination

### APPROACHES TO SOFTWARE ASSESSMENT

- Standard CMMI assessment (SCAMPI)
- CMM based appraisal for internal process improvement
- SPICE(ISO/IEC 15504)

ISO 9001:2000 for software Personal and Team Software Process Personal software process

PLANNING

HIGH LEVEL DESIGN

HIGH LEVEL DESIGN REVIEW

DEVELOPMENT

POSTMORTEM

### PERSONAL AND TEAM SOFTWARE PROCESS

Team software process Goal of TSP - Build self-directed teams - Motivate the teams - Acceptance of CMM level 5 behavior as normal to accelerate software process improvement - Provide improvement guidance to high maturity organization

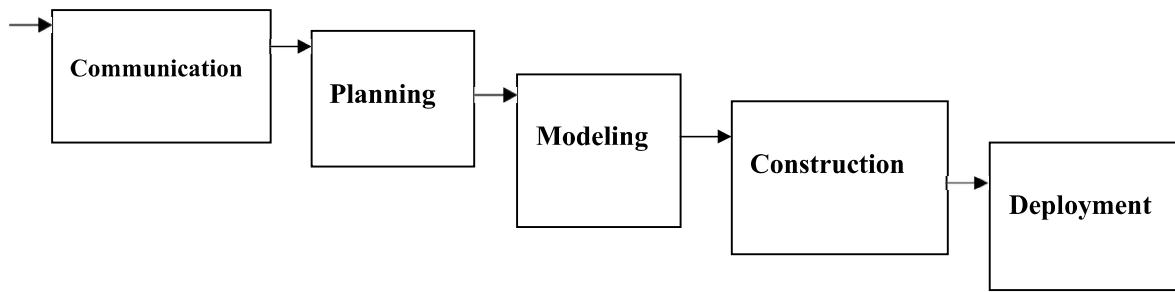
### PROCESS MODELS

- Help in the software development
- Guide the software team through a set of framework activities
- Process Models may be linear, incremental or evolutionary

### THE WATERFALL MODEL

- Used when requirements are well understood in the beginning
- Also called classic life cycle
- A systematic, sequential approach to Software development

- Begins with customer specification of Requirements and progresses through planning, modeling, construction and deployment.



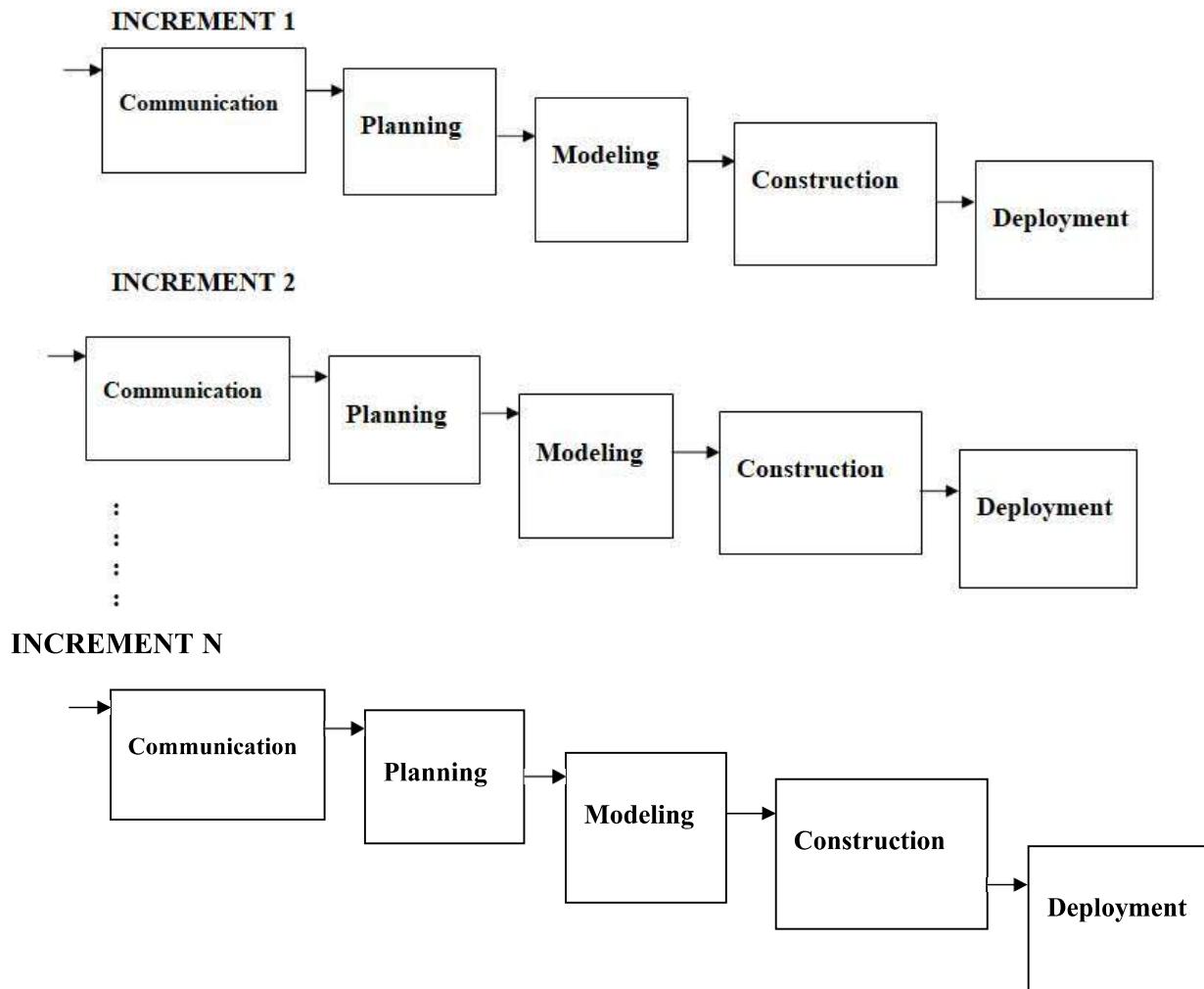
This Model suggests a systematic, sequential approach to SW development that begins at the systemlevel and progresses through analysis, design, code and testing

### **PROBLEMS IN WATERFALLMODEL**

- Real projects rarely follow the sequential flow since they are always iterative
- The model requires requirements to be explicitly spelled out in the beginning, which is oftendifficult
- A working model is not available until late in the project time plan

### **THE INCREMENTAL PROCESS MODEL**

- Linear sequential model is not suited for projects which are iterative in nature
- Incremental model suits such projects
- Used when initial requirements are reasonably well-defined and compelling need to provide limited functionality quickly
  - Functionality expanded further in later releases
  - Software is developed in increments
- Communication
- Planning
- Modeling
- Construction
- Deployment



- Software releases in increments
- 1st increment constitutes Core product
- Basic requirements are addressed
- Core product undergoes detailed evaluation by the customer
- As a result, plan is developed for the next increment. Plan addresses the modification of core product to better meet the needs of customer
- Process is repeated until the complete product is produced

### THE RAD (Rapid Application Development) MODEL

- An incremental software process model
- Having a short development cycle
- High-speed adoption of the waterfall model using a component based construction approach
- Creates a fully functional system within a very short span time of 60 to 90 days

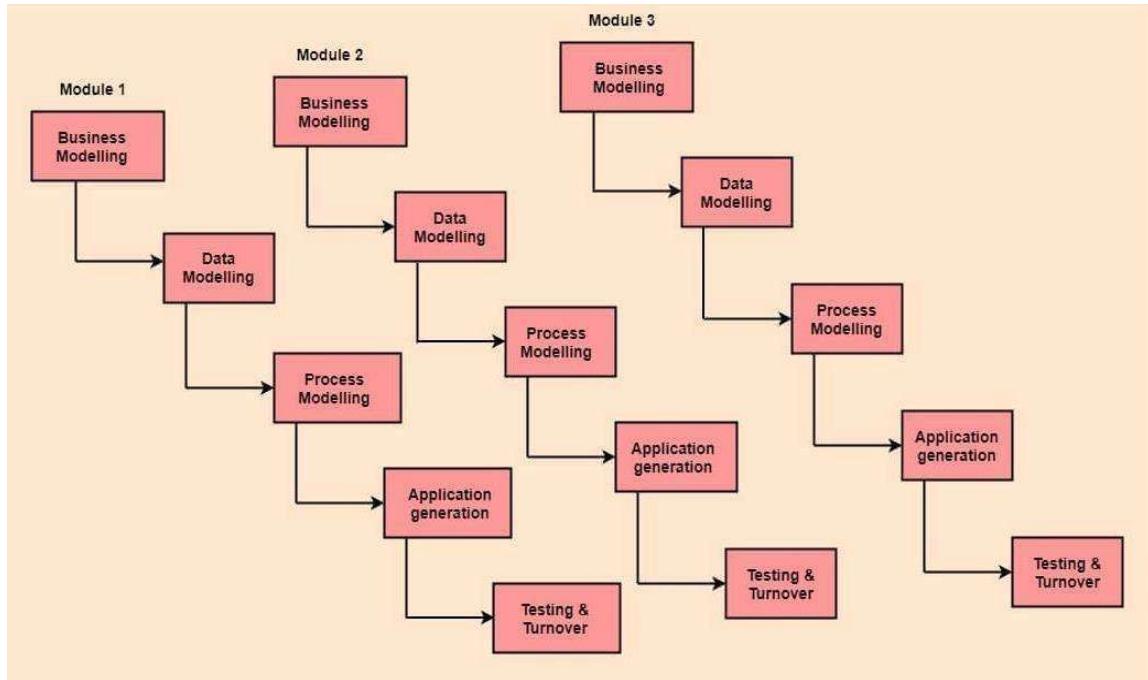
The RAD Model consists of the following phases:

**Communication** **Planning** **Construction** Component reuses automatic code generation testing  
Business modeling

Data modeling

Process modeling

**Deployment** integration delivery feedback



### THE RAD MODEL

- Multiple software teams work in parallel on different functions
- Modeling encompasses three major phases: Business modeling, Data modeling and processmodeling
- Construction uses reusable components, automatic code generation and testing

### Problems in RAD

- Requires a number of RAD teams
- Requires commitment from both developer and customer for rapid-fire completion of activities
- Requires modularity
- Not suited when technical risks are high

### EVOLUTIONARY PROCESSMODEL

- Software evolves over a period of time
- Business and product requirements often change as development proceeds making a straight-linepath to an end product unrealistic
- Evolutionary models are iterative and as such are applicable to modern day applications

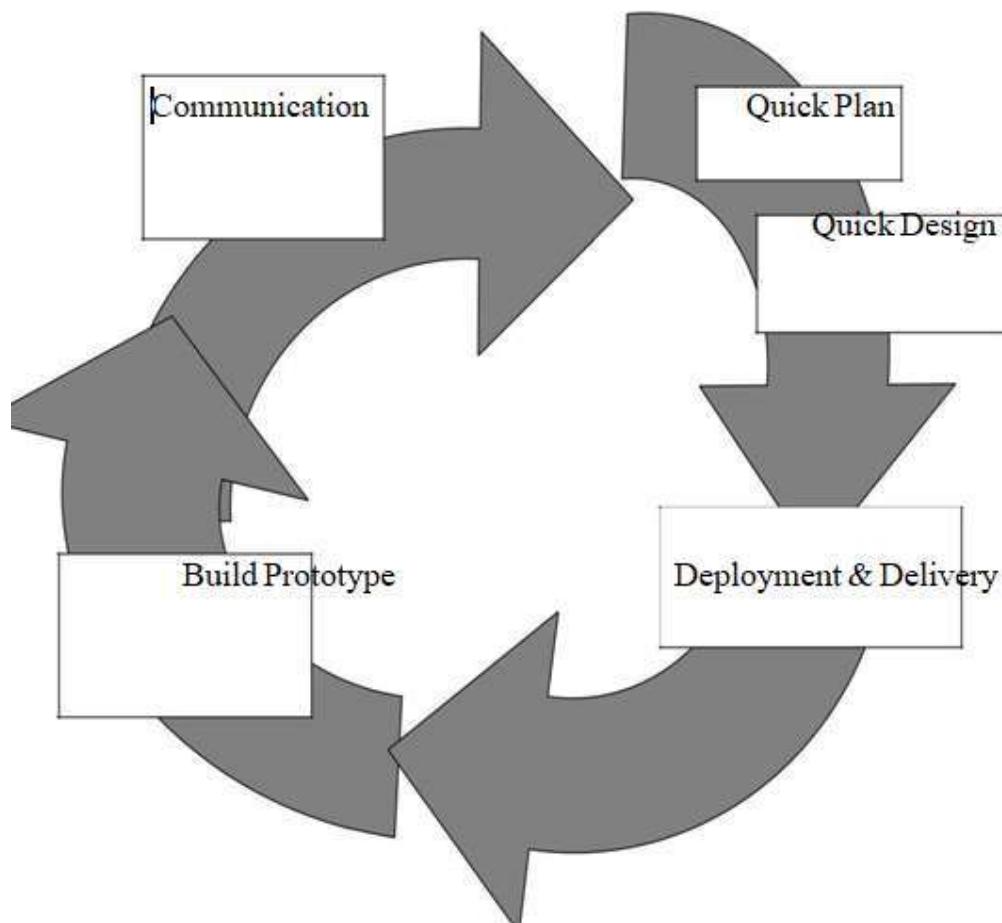
Types of evolutionary models

- Prototyping
- Spiral model
- Concurrent development model

### PROTOTYPING

- Mock up or model( throw away version) of a software product
- Used when customer defines a set of objective but does not identify input, output, or processing requirements
- Developer is not sure of:
- efficiency of an algorithm adaptability of an operating system

- human/machine interaction



### **STEPS IN PROTOTYPING**

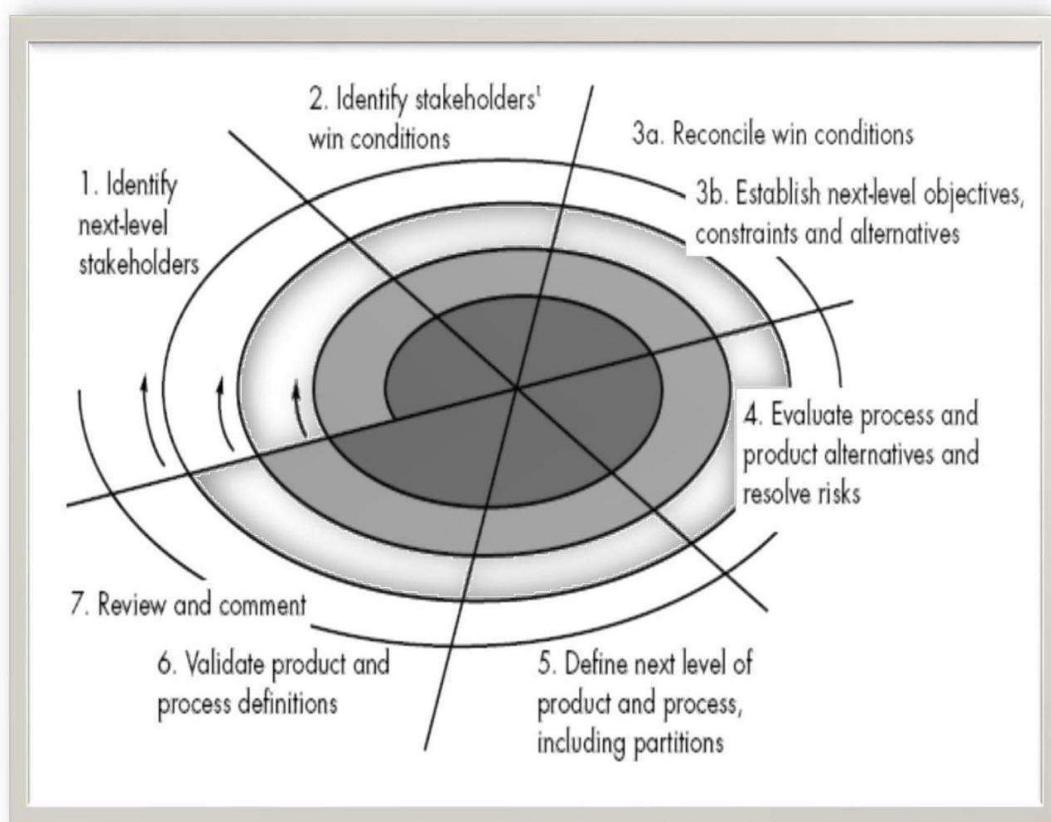
- Begins with requirement gathering
- Identify whatever requirements are known
- Outline areas where further definition is mandatory
- A quick design occur
- Quick design leads to the construction of prototype
- Prototype is evaluated by the customer
- Requirements are refined
- Prototype is turned to satisfy the needs of customer
- 

### **LIMITATIONS OF PROTOTYPING**

- In a rush to get it working, overall software quality or long term maintainability are generally overlooked
- Use of inappropriate OS or PL
- Use of inefficient algorithm

### **THE SPIRAL MODEL**

An evolutionary model which combines the best feature of the classical life cycle and the iterative nature of prototype model. Include new element : Risk element. Starts in middle and continually visits the basic tasks of communication, planning, modeling, construction and deployment



## THE SPIRAL MODEL

- Realistic approach to the development of large scale system and software
- Software evolves as process progresses
- Better understanding between developer and customer
- The first circuit might result in the development of a product specification
- Subsequent circuits develop a prototype
- sophisticated version of software

## THE CONCURRENT DEVELOPMENT MODEL

- The concurrent development model also called concurrent engineering
- Constitutes a series of framework activities, software engineering action, tasks and their associated states
- All activities exist concurrently but reside in different states
- Applicable to all types of software development
- Event generated at one point in the process trigger transitions among the states

## A FINAL COMMENT ON EVOLUTIONARY PROCESS

- Difficult in project planning
- Speed of evolution is not known

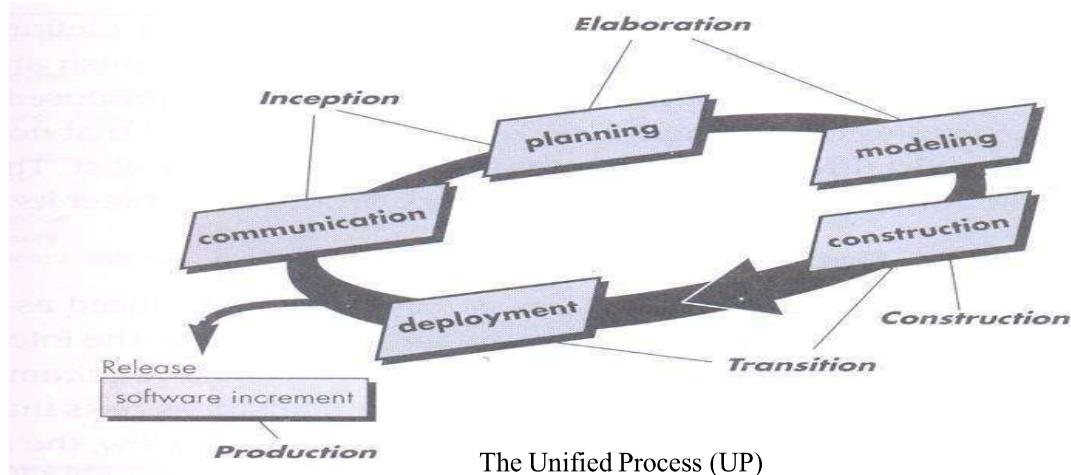
- Does not focus on flexibility and extensibility (more emphasis on high quality)
- Requirement is balance between high quality and flexibility and extensibility

## **THE UNIFIED PROCESS**

Evolved by Rumbaugh, Booch, Jacobson. Combines the best features their OO models. Adopts additional features proposed by other experts. Resulted in Unified Modeling Language (UML). Unified process developed Rumbaugh and Booch. A framework for Object-Oriented Software Engineering using UML

### **PHASES OF UNIFIED PROCESS**

- INCEPTION PHASE
- ELABORATION PHASE
- CONSTRUCTION PHASE
- TRANSITION PHASE



### **UNIFIED PROCESS WORK PRODUCTS**

Tasks which are required to be completed during different phases

#### **1. Inception Phase**

- \*Vision document
- \*Initial Use-Case model
- \*Initial Risk assessment
- \*Project Plan

#### **2. Elaboration Phase**

- \*Use-Case model
- \*Analysis model
- \*Software Architecture description
- \*Preliminary design model
- \*Preliminary model

#### **3. Construction Phase**

- \*Design model

- \*System components
- \*Test plan and procedure
- \*Test cases
- \*Manual

#### **4. Transition Phase**

- \*Delivered software increment
- \*Beta test results
- \*General user feedback