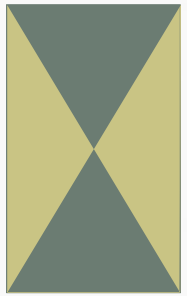
CHAPTER 1:
SOFTWARE AND SOFTWARE ENGINEERING



TOPICS COVERED IN THIS CHAPTER:

- A definition of software
- Dual role of software
- Differences between hardware and software
- Changing nature of software
- Software myths

WHAT IS SOFTWARE ENGINEERING?

- Consists of two terms:
  - Software
  - Engineering

WHAT IS SOFTWARE ENGINEERING?

- **SOFTWARE:**
  - It is a program/set of programs containing instructions which provide desired features, functionality and performance.
  - It also comprises of **Data structures** that enable the programs to manipulate information.
- **ENGINEERING:**
  - The process of designing and building something that serves a particular purpose.
  - Role is to provide systems and products that enhance the material aspects of human life.

WHAT IS SOFTWARE ENGINEERING?

Definition:

“A systematic approach to the development, operation and maintenance of the desired software.”

DUAL ROLE OF SOFTWARE:

- As a **Product**
  - Delivers computing potential of a computer hardware
  - Enables the H/W to deliver the expected functionality
  - Acts as an Information Transformer (produces, manages, acquires, modifies, display the information)
    - Business Information
      - Ex: Query retrieval SW based on market data
    - Personal Information
      - Ex: Individual's financial transactions

DUAL ROLE OF SOFTWARE:

- As a **Vehicle** for delivering a product
  - Helps in creation and control of other programs.
    - Act as a control of the computer(e.g. Operating Systems)
    - Communications of information(e.g. Networking Software)

What is work product?

- **Software Engineer:**
  - The set of programs, the content (data) along with documentation that is a part of SW.
- **User/Customer:**
  - The functionality delivered by the SW that improves user experience.

Why Software Engineering Important?

- Helps to build complex system in a timely manner
- Ensures high quality of products
- Focuses on
  - Quality
    - Functional: Degree to which correct SW is produced
    - Non-functional: features other than function of a SW
  - Maintainability
    - It should be easily enhanced whenever required and adapted to changing requirements.

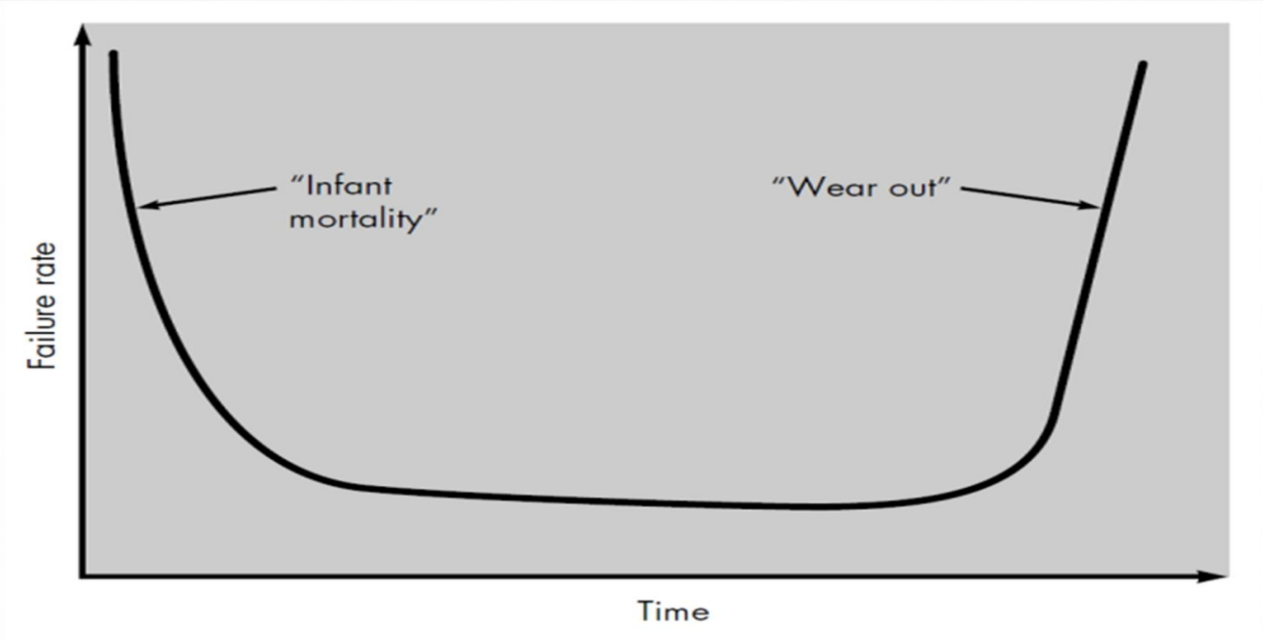
COMPARISON BETWEEN SOFTWARE AND HARDWARE:

SOFTWARE	HARDWARE
● Software is developed or engineered	● HW is manufactured
● SW doesn't wear out	● HW wear out with time
● SW is a logical unit ○ Intangible in nature	● HW is a physical unit ● Tangible in nature
● Problem Statement ○ Initially, may be not complete & clear	● Clearly specified at the beginning
● Requirements ○ May change with time	● Fixed before manufacturing

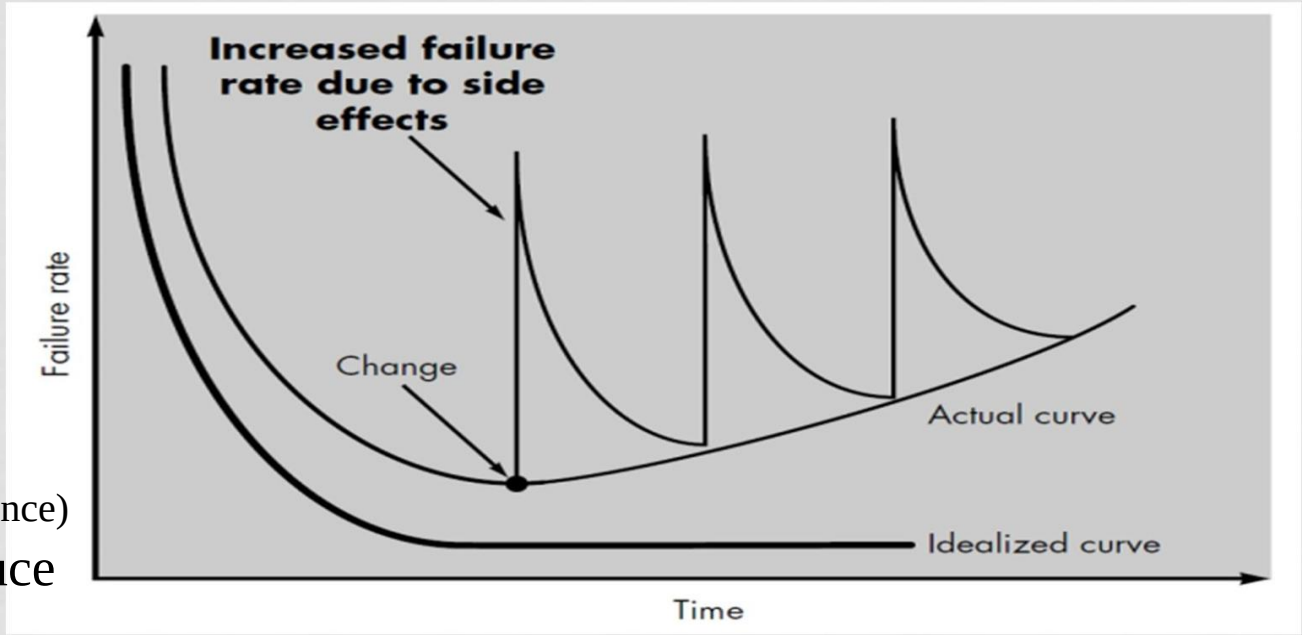
COMPARISON BETWEEN SOFTWARE AND HARDWARE:

SOFTWARE	HARDWARE
● Multiple copies ○ At low cost	● At higher cost
● No software spare parts ○ Dependent modules	● Spare parts for HW exists ○ Independent parts

HARDWARE FAILURE CURVE:



SOFTWARE FAILURE CURVE:



(maintenance)

Introduce new defects

SOFTWARE APPLICATION DOMAIN

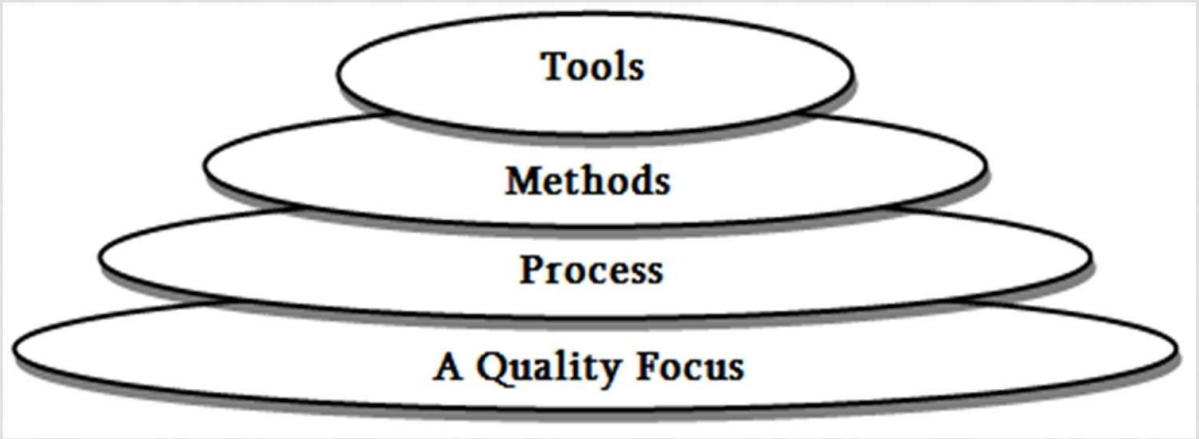
- System Software
- Application Software
- Engineering/Scientific Software
- Embedded Software
- Artificial Intelligence Software
- Web/Mobile Application Software

TODAY’S SOFTWARE DEVELOPMENT:

- Challenges:
  - Development of large & complex systems
  - Software systems must fulfill the requirements of a client
  - Number of persons involved in the development>1
  - Software systems are expected to live long and be used by many people
  - Increased quality demands on software products
  - High cost and time pressure
  - Difficulty in measuring Software development progress

LAYERED TECHNOLOGY:

- Software engineering is a layered technology.



CONT…

- SE comprises of a process, a set of methods for managing and developing a SW and a collection of tools.
- Quality Focus:
  - Degree to which a **correct** SW has been developed
  - The degree to which a system meets specified requirements, customer needs and expectations.
  - **Maintainability**
  - **Reusability**
  - **Reliability** and **Portability**
  - SE must rest on an organizational commitment to quality
  - Total quality management raise a continuous process improvement activity.

CONT…

- Process:
  - What to do?
  - Deals with Activities, Actions and Tasks.
  - A framework that must be established for effective delivery of SW.
  - Timely development of the SW
  - Management and control of SW projects.
  - How to do?

CONT…

- Methods:
  - Deals with ‘How to’ implement
    - Communication
    - Requirement and design modeling analysis
    - Testing and Support
  - Provide the technical way to building software.

CONT…

- Tools:
  - Provides environment
  - SE tools provide automated or semi-automated support for the process and the methods.
  - Helping hand of process
  - Used for code, design, test, etc..

THE SOFTWARE PROCESS:

- It is a collection of activities, actions and tasks that are performed when product is to be created.
- An **activity** is to achieve a broad objective (communication with client) and is applied regardless of the application domain, size of project and complexity of effort.
- An **action** (design) encompasses a set of tasks that produce a product (design model).
- A **task** focuses on small but well-defined objective that produces a tangible outcome (unit testing).

PROCESS FRAMEWORK ACTIVITIES:

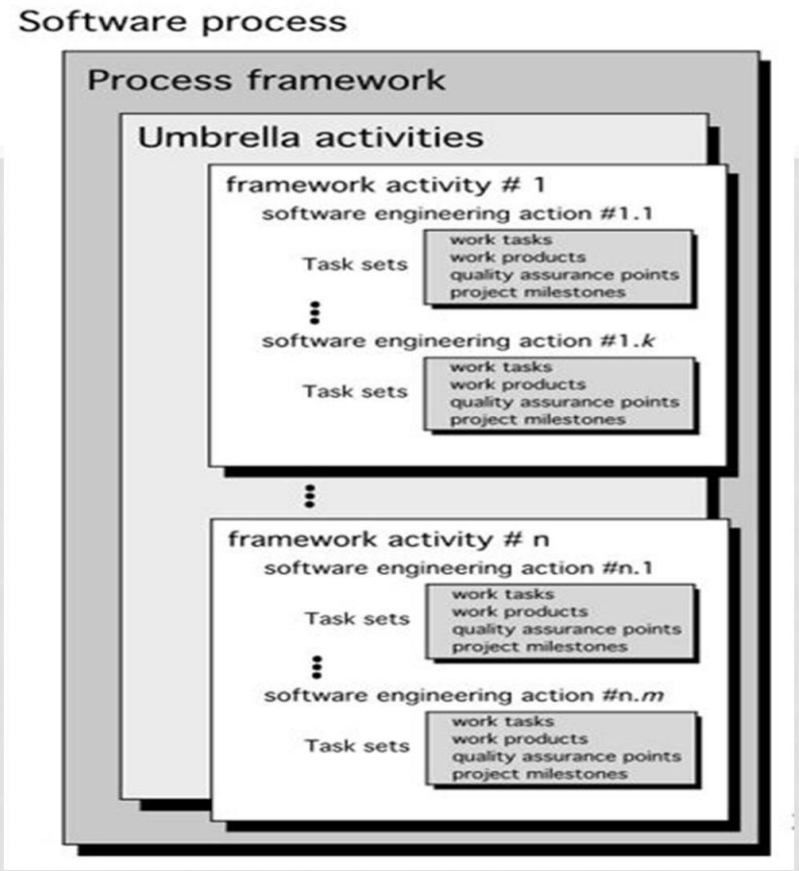
- Communication
  - Before the technical work, it is important to communicate with the customer.
  - Understand objective of the project.
  - Gather the requirements to define features and functions of the software.
- Planning
  - It creates a map that helps to guide the team.
  - It defines SE work by describing technical tasks, the risks, required resources, work schedule.

CONT…

- Modeling
  - To better understand software requirements and the design to achieve requirements.
- Construction
  - It combines code generation and the testing that is required to find errors.
- Deployment
  - Software is delivered to the customer who use the product and provide feedback.

PROCESS MODEL:

- Each framework activity
  - Set of actions
    - Set of tasks
- Task
  - Work task
  - Requirement gathering
  - Quality assurance points
  - Project milestones



FRAMEWORK ACTIVITY:

- For small software project:
  - Requested by one person
  - Straightforward requirements
  - Communication activity carried on phone call (phone conversation)
  - Work tasks
    - Communication via telephone
    - Discuss requirements and take notes
    - Organize notes into brief
    - E-mail to client for review and approval

CONT…

- For complex and large software project:
  - Many stakeholders
  - Each have different requirements
  - Communication activity (six actions)
    - Inception
      - To establish basic understanding of problem, proper solution, effective communication with client and software team.
    - Elicitation
      - Objectives for system
      - How the system is to be used on day-to-day basis
    - Elaboration
      - Expand and refined information which is gathered during inception and elicitation

CONT…

- Negotiation
  - Discuss about priority of requirements
  - Assess cost and risk
  - Requirements are eliminated, combined, modified
- Specification
  - Written document
  - Graphical models
- Validation
  - Assess quality

Example: Task set

- **Requirements Gathering (For small project):**
  - Make a list of stakeholders for the project.
  - Invite all stakeholders to an informal meeting.
  - Ask each stakeholders to make a list of features and functions required.
  - Discuss requirements and build a final list.
  - Prioritize requirements.
  - Note areas of uncertainty

Example: Task set

- **Requirements Gathering (For large, complex project):**
  - Make a list of stakeholders for the project.
  - Interview each stakeholder to determine needs.
  - Make a preliminary list of features and functions required.
  - Conduct meetings.
  - Produce informal user scenarios as a part of each meetings.
  - Refine user scenarios based on feedback and build revised list.
  - Prioritize requirements.
  - Note constraints and restrictions.
  - Discuss methods for validating the system.

UMBRELLA ACTIVITIES:

- Umbrella Activities are those activities to be performed through the entire Software Process.
- It helps a software team to manage and control progress, quality, change, risk, etc..
- It includes:
  - Software project tracking and control
    - Assess progress against the project plan and take necessary action.
  - Risk management
    - Assess the risk that affect the outcome of product.
  - Software quality assurance

CONT…

- Technical reviews
- Software configuration management
  - Manages the effects of change throughout the software process.
- Reusability management
  - Manage the reusable components
- Work product preparation and production
  - Encompasses the activities required to create work product such as models, documents, etc.

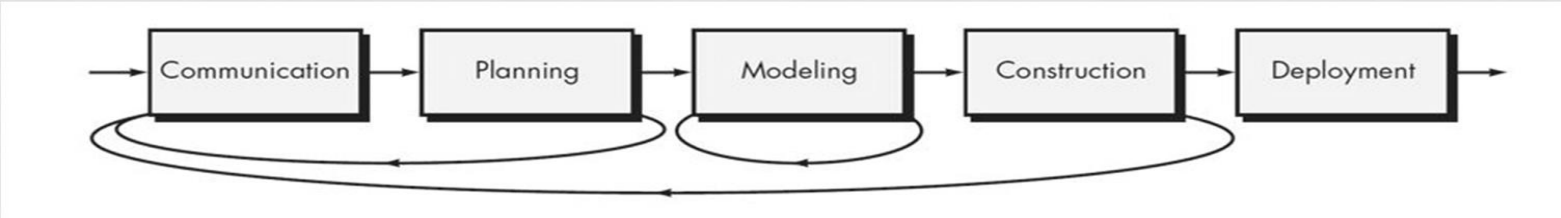
PROCESS FLOW:

- Linear process flow
- Iterative process flow
- Evolutionary process flow
  - Circular manner
- Parallel process

CONT…

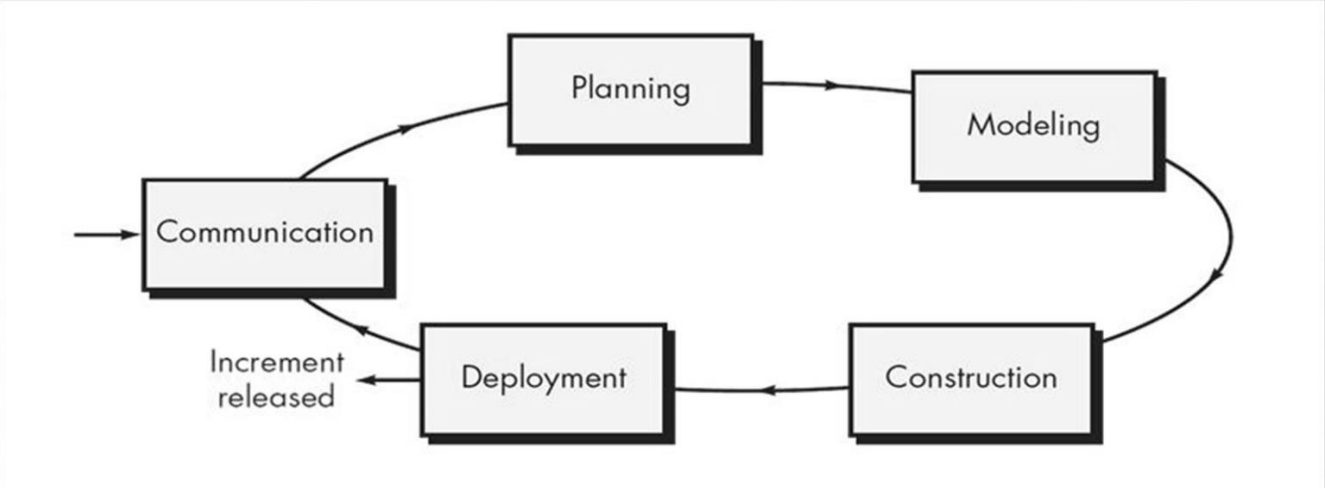
- Linear process flow

- Iterative process flow



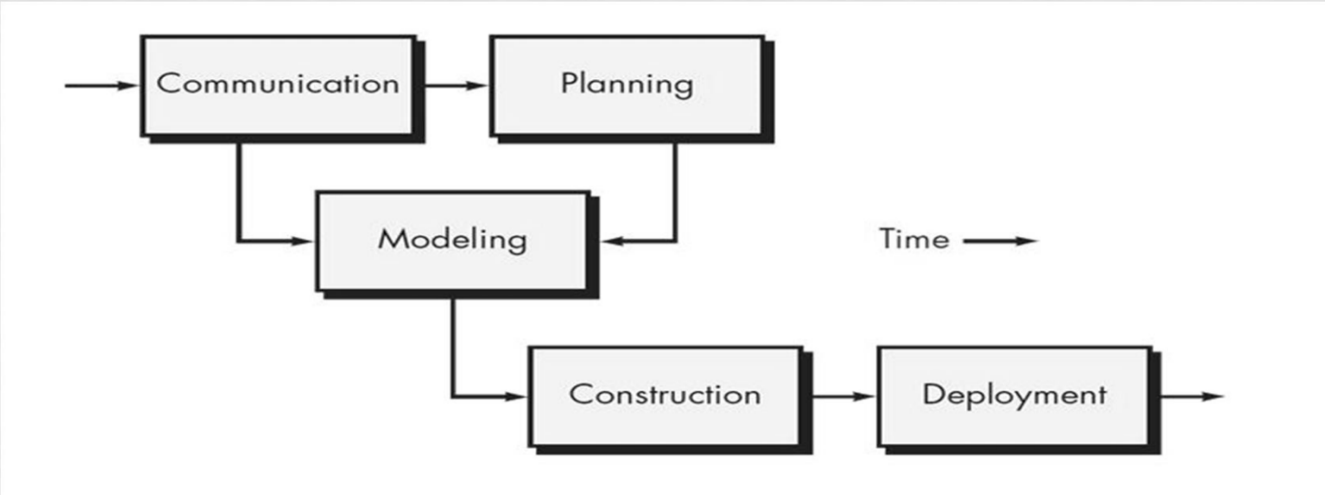
CONT…

- Evolutionary process flow (Circular Manner)



CONT…

- Parallel process



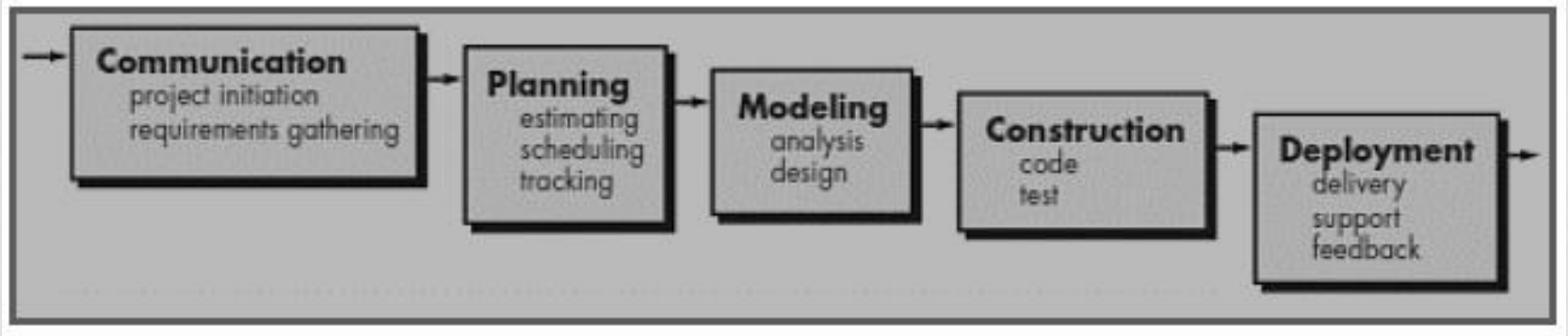
PROCESS MODELS:

“The process followed in the development of the SW depends upon the life cycle model chosen for development.”

- Waterfall process model
- Incremental process model
  - RAD model (Rapid Application Development)
- Evolutionary process model
  - Prototype model
  - Spiral model
- Concurrent process model

WATERFALL MODEL:

- Classic life cycle, suggests a systematic, sequential approach to software development.
- It begins with customer specification of requirements and progress through planning, modeling, construction and deployment.
- Each phase executed sequentially without overlap.



CONT…

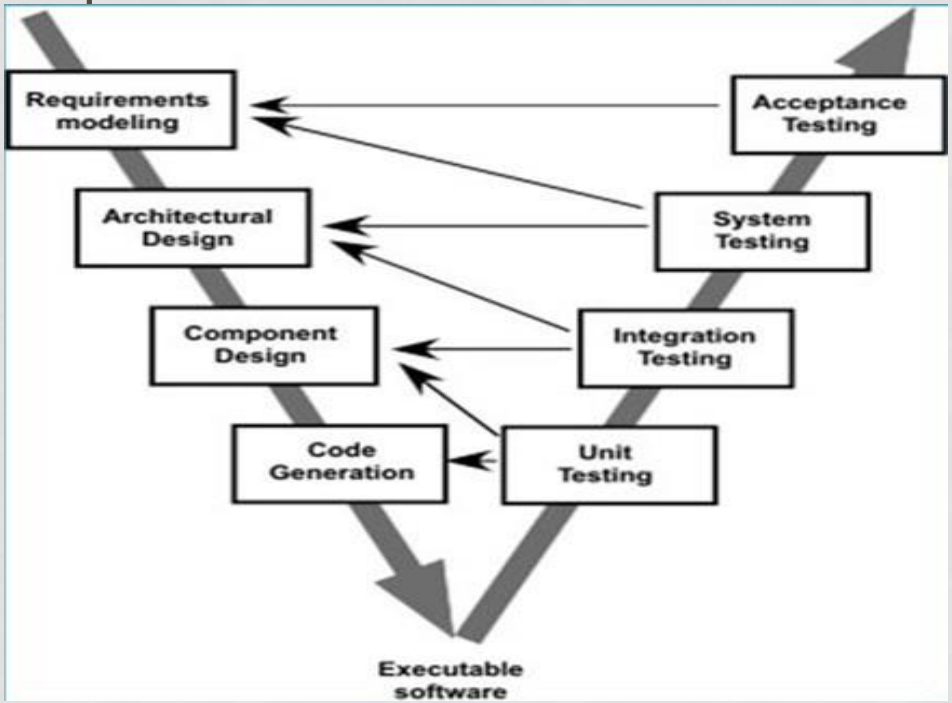
- Limitations:
  - The nature of requirements will not change very much during development
  - Implies that complete a given stage before moving on to the next stage
  - Does not account for the fact that requirements constantly change
  - Customer can’t use anything until the entire system is complete
  - Surprises at the end are very expensive
    - No assessment of risk.
    - Doesn’t handle an unexpected risks.
  - Some teams sit ideal for other teams to finish
  - This is only appropriate when the requirements are well-understood

WHEN TO USE WATERFALL MODEL?

- Requirements are very well known, clear and fixed
- Product definition is stable
- Technology is understood
- Sufficient resources with required expertise are available
- The project is small
- Can be used by an organization
  - That has an experience in developing a particular SW.
  - When it wants to build a new SW based on an existing SW.

V-MODEL:

- Variation in the representation of the waterfall model



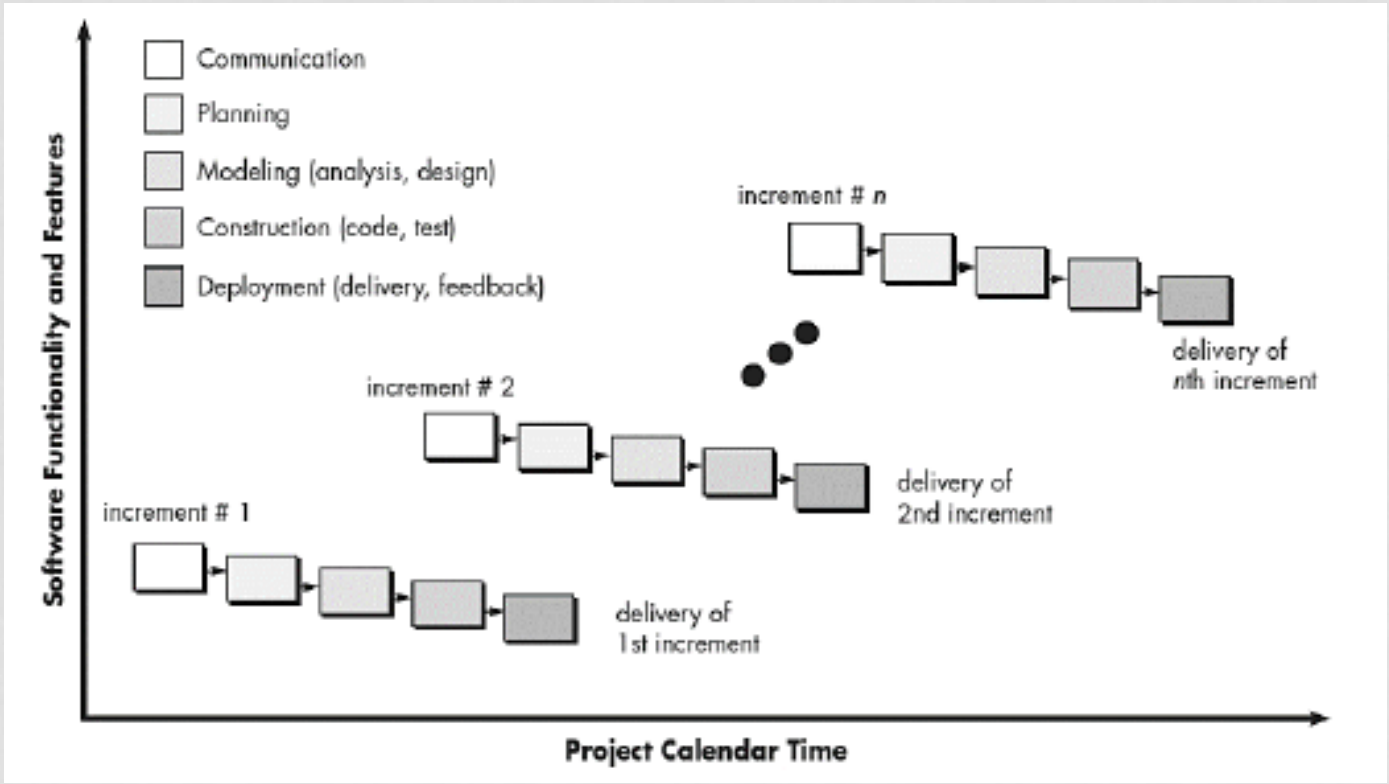
CONT…

- As a software team moves down the left side of the V, basic problem requirements are refined into progressively more detailed, technical representations of the problem and its solution.
- After the generation of code, team moves to the right side of the V.
- Perform series of tests that validate each of the model created in the all phase of the left side.
- It provides a way of visualizing how verification and validation actions are applied.

INCREMENTAL MODEL:

- It combines elements of linear and parallel process flows.
- Each linear sequence produces increments of the software.
- Each subsequent release of the module adds function to the previous release.
- **Ex:** word processing software developed using this model.
  - First increment: file management, editing, document production
  - Second increment: more sophisticated editing, document production
  - Third increment: spelling and grammar checking

CONT…


Communication
Planning
Modeling (analysis, design)
Construction (code, test)
Deployment (delivery, feedback)
increment # n
delivery of
nth increment
increment # 2
delivery of
2nd increment
increment # 1
delivery of
1st increment
Software Functionality and Features
Project Calendar Time

ADVANTAGES:

- Generates working software quickly during the software life cycle.
- More flexible and less costly to change scope and requirements.
- Easier to test during small iteration.
- Customer can respond to each built.
- Easier to manage risk because mostly risk handled during iteration.

DISADVANTAGES:

- Needs good planning and design.
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
- Total cost is higher than waterfall.

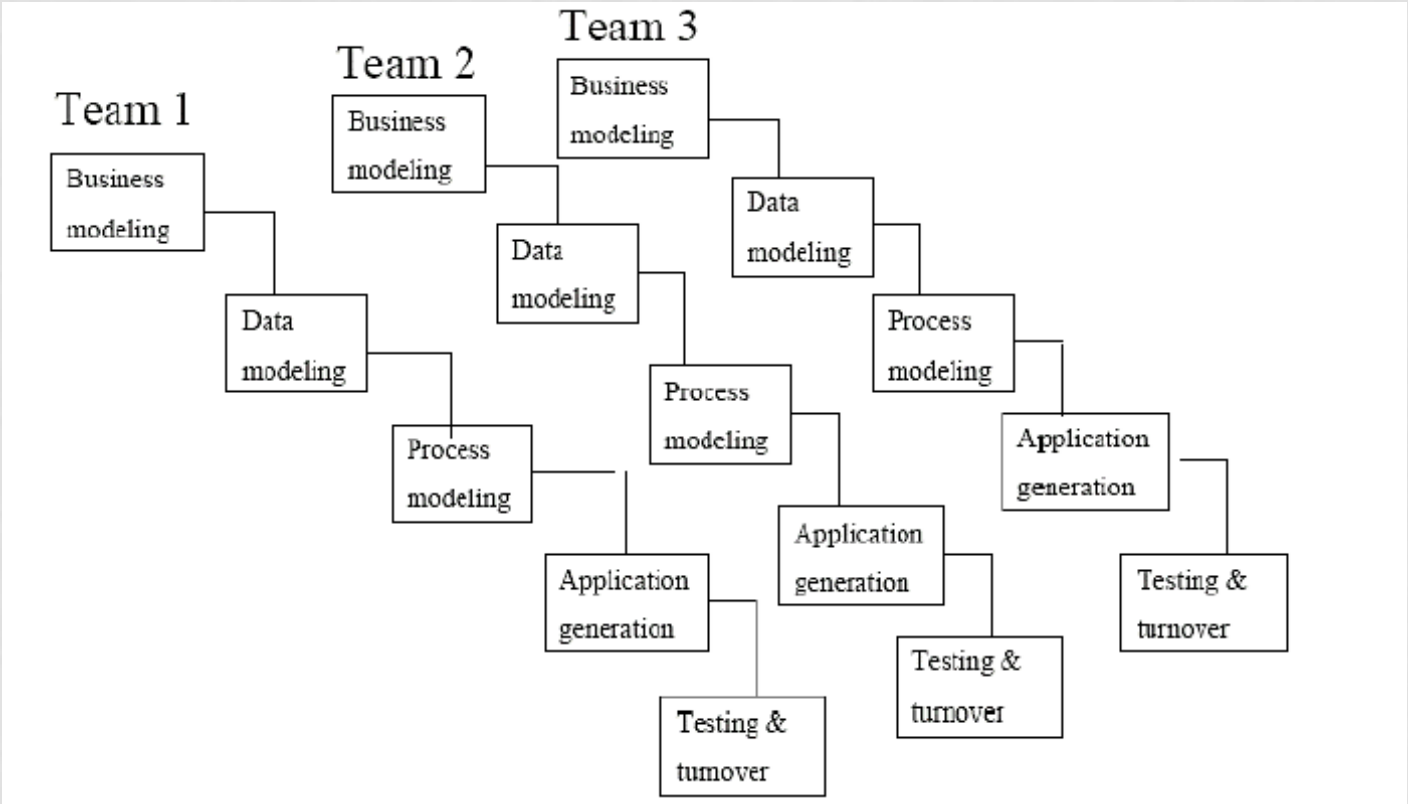
WHEN TO USE?

- This model can be used when the requirements of the complete system are clearly defined and understood.
- Major requirements must be defined so that some details can evolve with time.
- There is a need to get a product to the market early.
- Resources with needed skill set are not available.

RAD MODEL:

- Rapid Application Development
- In this model the components or functions are developed in parallel as if they were mini projects.
- The developments are time boxed, delivered and then assembled into a working prototype.
- This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.

CONT…


Team 1
Business modeling
Data modeling
Process modeling
Application generation
Testing & turnover
Team 2
Business modeling
Data modeling
Process modeling
Application generation
Testing & turnover
Team 3
Business modeling
Data modeling
Process modeling
Application generation
Testing & turnover

CONT…

- The phases in the RAD model are:
- **Business modeling:** The information flow is identified between various business functions.
- **Data modeling:** Information gathered from business modeling is used to define data objects that are needed for the business.
- **Process modeling:** Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective.
- **Application generation:** Automated tools are used to convert process models into code and the actual system.
- **Testing and turnover:** Test new components and all the interfaces

ADVANTAGES:

- Reduced development time.
- Increases reusability of components
- Quick initial reviews occur
- Encourages customer feedback
- Integration from very beginning solves a lot of integration issues.

DISADVANTAGES:

- Depends on strong team and individual performances for identifying business requirements.
- Requires highly skilled developers/designers.
- High dependency on modeling skills.
- Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.

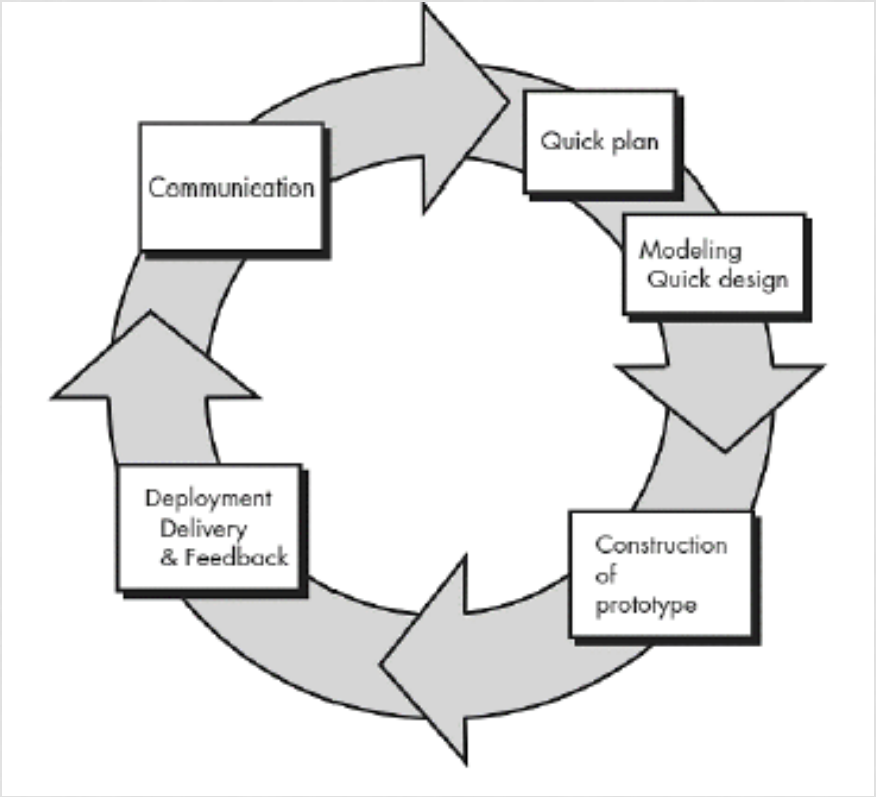
WHEN TO USE?

- It should be used when there is a need to create a system that can be modularized in 2-3 months of time.
- Used if there is high availability of designers for modeling and the budget is high enough to afford their cost along with the cost of automated code generating tools.
- When customer requirements are same as another application.

PROTOTYPE MODEL:

- The basic idea of this model is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements and this prototype is developed based on the currently known requirements.
- By using this prototype, the client can get an “actual feel” of the system by the interactions with prototype.
- It is more useful idea for **complicated and large systems** for which there is no manual process or existing system to help determining the requirements.
- The prototype are usually not complete systems and many of the details are not built in the prototype.
- The goal is to provide a system with overall functionality.

CONT…


Communication
Quick plan
Modeling
Quick design
Construction
of
prototype
Deployment
Delivery
& Feedback

ADVANTAGES:

- Users are actively involved in the development.
- Users get a better understanding of the system being developed by using working model of the system is provided.
- Errors can be detected much earlier.
- Quicker user feedback is available for better solutions.
- Missing functionality can be identified easily.
- Confusing or difficult functions can be identified.

DISADVANTAGES:

- Practically, this model may increase the complexity of the system as scope of the system may expand beyond original plans.

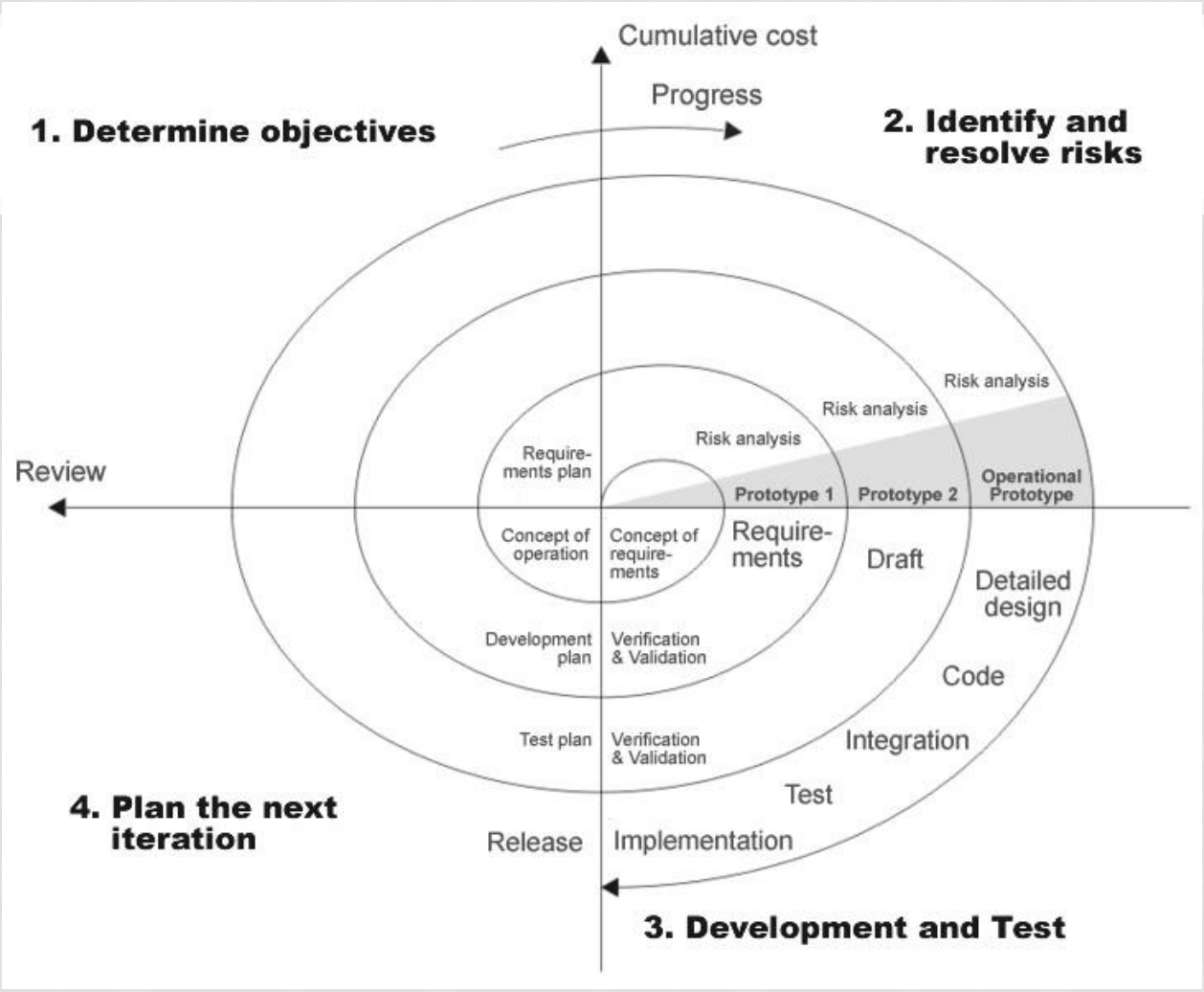
WHEN TO USE?

- Used when the desired system needs to have a lot of interaction with the end users.
- Generally, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model.
- Prototyping ensures that the end users constantly work with the system and provide a feedback.

SPIRAL MODEL:

- It provides potential for rapid development of increasingly more complete versions of the software.
- It has 2 features:
  - Cyclic approach for incrementally growing an implementation while decreasing risk.
  - Set of anchor point milestones for ensuring stakeholder commitment to feasible and satisfactory system.
- **Cost and schedule are adjusted** based on feedback derived from the customer after delivery.
- The project manager adjusts the planned number of iterations required to complete the software.

CONT…


Cumulative cost
Progress
1. Determine objectives
2. Identify and resolve risks
Review
Require-ments plan
Risk analysis
Risk analysis
Risk analysis
Prototype 1
Prototype 2
Operational Prototype
Concept of operation
Concept of require-ments
Require-ments
Draft
Detailed design
Development plan
Verification & Validation
Code
Test plan
Verification & Validation
Integration
Test
4. Plan the next iteration
Release
Implementation
3. Development and Test

ADVANTAGES:

- High amount of risk analysis so that avoidance of Risk is enhanced.
- Good for **large and mission-critical** projects.
- **Strong approval** and documentation control.
- Additional functionality can be added at a later date.

DISADVANTAGES:

- Can be a costly model to use.
- Risk analysis requires highly specific expertise.
- Project’s success is highly dependent on the risk analysis phase.

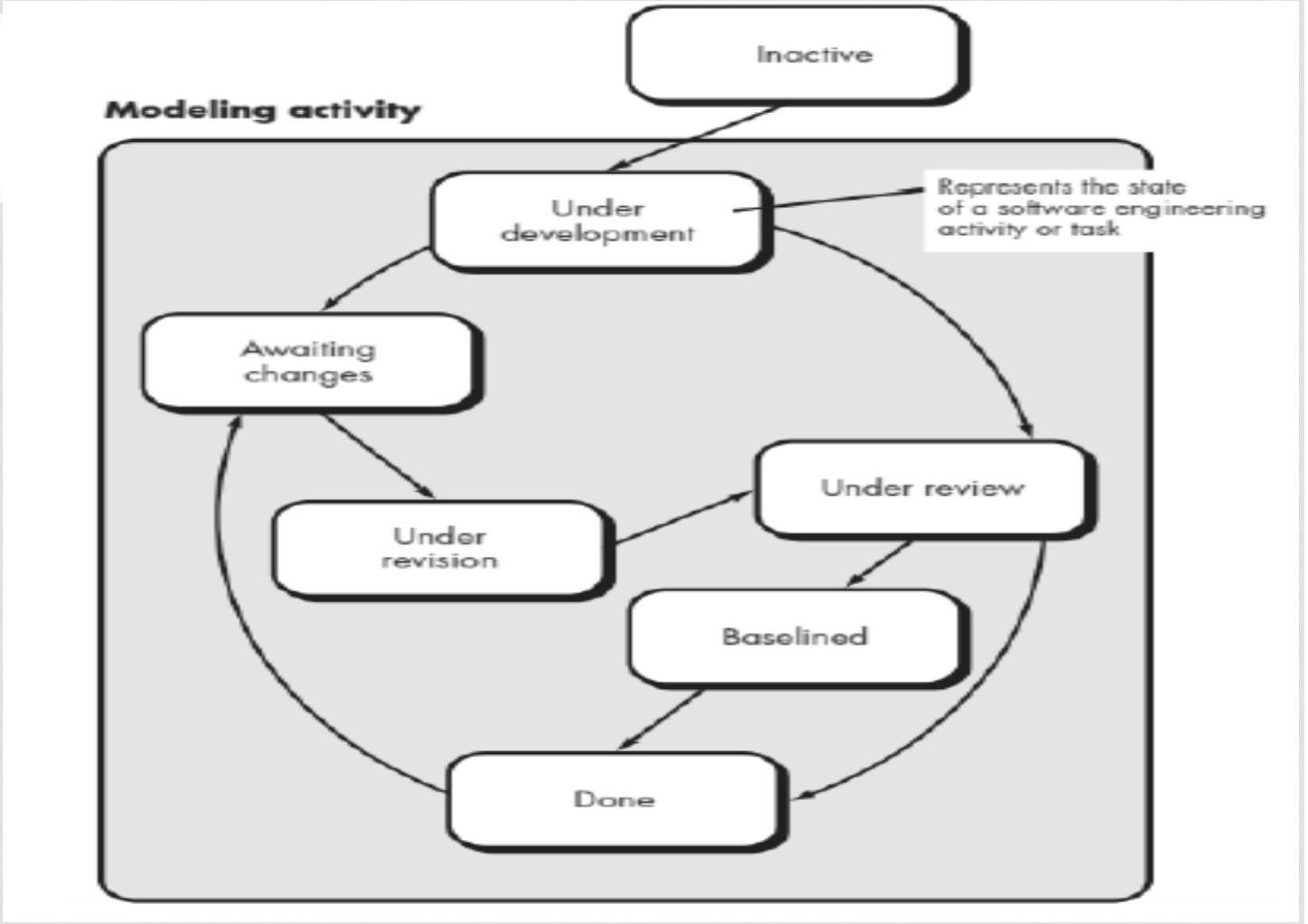
WHEN TO USE?

- When costs and risk evaluation is important
- For medium to high-risk projects
- Users are unsure of their needs
- Requirements are complex
- Significant changes are expected (research and exploration)

CONCURRENT MODEL:

- The concurrent development model, sometimes called concurrent engineering, allows a software team to represent iterative and concurrent elements of any of the process models.
- Ex:
  - For modeling activity concurrent actions:
    - Prototyping
    - Analysis
    - Design

CONT…


Inactive
Modeling activity
Under development
Represents the state of a software engineering activity or task
Awaiting changes
Under review
Under revision
Baselined
Done

CONCURRENT MODEL:

- The activities in any one of the states at given time.
- All activities exist concurrently but reside in different states.
- It defines a series of events that trigger transitions from state to state for each activity, action or task.
- Applicable to all type of software development and provides an accurate picture of the current state.

AGILE MODEL:

- Agile SDLC model is a combination of **iterative and incremental process** models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
- Agile methods break the product into small incremental builds.
- Every iteration involves functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing.

ADVANTAGES:

- Customer satisfaction by rapid, continuous delivery of useful software.
- Customers, developers and testers constantly interact with each other.
- Daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed.

DISADVANTAGES:

- In case of some software, it is difficult to assess the effort required at the beginning of the software development life cycle.
- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process.

PRODUCT AND PROCESS:

- If the process is weak, the end product will undoubtedly suffer.
- People derive as much (or more) satisfaction from the creative process as they do from the end product.
  - An artist enjoys the brush strokes as much as the framed result.
- As creative software professional, you should also derive as much satisfaction from the process as the end product.
- The duality of product and process is one important element in keeping creative people engaged as software engineering continues to evolve.

VALIDATION VS. VERIFICATION:

Verification	Validation
● It is a static practice of verifying documents, design, code and program.	● It is a dynamic mechanism of validating and testing the actual product.
● It **does not involve** executing the code.	● It always **involves** executing the code.
● It is **human based** checking of documents and files.	● It is **computer based** execution of program.
● It uses methods like **inspections, reviews, walkthroughs**, etc.	● It uses methods like **black box (functional) testing, white box (structural) testing**, etc.
● It is to check whether the software **conforms to specifications**.	● It is to check whether software **meets the customer expectations and requirements**.

VALIDATION VS. VERIFICATION:

Verification	Validation
● Target is **requirements specification**, application and software architecture, complete design and database design etc.	● Target is **actual product**-a unit, a module, integrated modules and effective final product.
● It is done by **QA team** to ensure that the software is as per the specifications in the SRS document.	● It is carried out with the involvement of **testing team**.
● It generally comes before validation.	● It generally follows after verification.

Exercise:

1. Which one of the following models is not suitable for accommodating any change?
    a) Incremental Model
    b) Prototyping Model
    c) RAD Model
    d) Waterfall Model

Answer: d) Waterfall Model

Exercise:

2. Which model is more useful for **complicated and large systems** for which there is no existing system to determining the requirements.

Answer: d) Prototype Model

Exercise:

4. If you are a lead developer of a software company and you are asked to submit a project/product within a stipulated time-frame with no cost barriers, which model would you select?
    a) Waterfall
    b) Spiral
    c) RAD
    d) Incremental

Answer: c) RAD

Exercise:

5. Which is the good approach when a working core product is required quickly?

Answer: Incremental model

6. Which is the useful approach when a customer cannot define requirements clearly?

Answer: Prototyping model

Exercise:

Which of the following life cycle model can be chosen if the development team has less experience on similar projects?

a) Spiral
b) Waterfall
c) RAD
d) Iterative Enhancement Model

Answer: a) Spiral

Exercise:

Which two models doesn’t allow defining requirements early in the cycle?

a) Waterfall & RAD

b) Prototyping & Spiral

c) Prototyping & RAD

d) Waterfall & Spiral

Answer: b) Prototyping & Spiral

Exercise:

A company is developing an advance version of their current software available in the market, what model approach would they prefer ?

Answer: Both RAD & Incremental Model

Which model is not applicable for maintenance projects?

Answer: Waterfall Model

Exercise:

You have been asked to develop a small application that analyses each course offered by a university and reports the average grade obtained in the course (for a given term). What software model(s) would you choose and why?

Answer: Waterfall model

Exercise:

You have been appointed a project manager within an information systems organization. Your job is to build an application that is quite similar to others your team has built, although this one is larger and more complex. Requirements have been thoroughly documented by the customer and add only new requirements. What software process model(s) would you choose and why? Explain in detail.

Answer: Incremental model

Exercise:

Let's say there is a solution company that basically which were expert in installing this particular HR management systems in big retailers. So let's say company X need to install a well known HR management system at a big retailer's headquarter. And they have done this install many times and similar retailers, big retailers before. So the question is, what model will fit best in this situation?

Answer: Waterfall model