Object Oriented Software Development

Chapter 1: Introduction

- Software:
  - is a generic term for organized collections of computer data and instructions.
  - Two types:
    - System Software:
      - is responsible for controlling, integrating, and managing the individual hardware components of a computer system.
      - E.g. operating system
    - Application Software:
      - is used to accomplish specific tasks other than just running the computer system.
      - E.g. Microsoft Office

Introduction….

- Software Engineering:
  - is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures.
  - The outcome of software engineering is an efficient and reliable software product.
  - is the process of solving customer's problem by the systematic development and evolution of large, high quality software systems within cost, time and other constraints.

Introduction…..

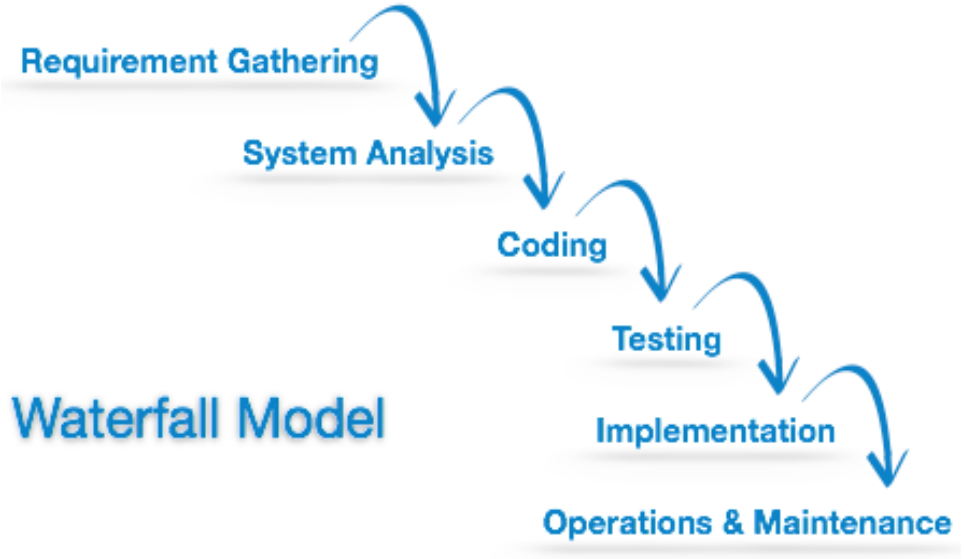
- Software Development Life Cycle (SDLC):
  - is a well-defined, structured sequence of stages in software engineering to develop the intended software product.

Introduction……

- Software Development Paradigm:
  - helps developer to select a strategy to develop the software.
  - A software development paradigm has its own set of tools, methods and procedures, which are expressed clearly and defines software development life cycle.
  - A few of software development paradigms or process models are defined as follows:
    - Waterfall Model
    - Iterative Model
    - Spiral Model
    - V Model
    - Big Bang Model

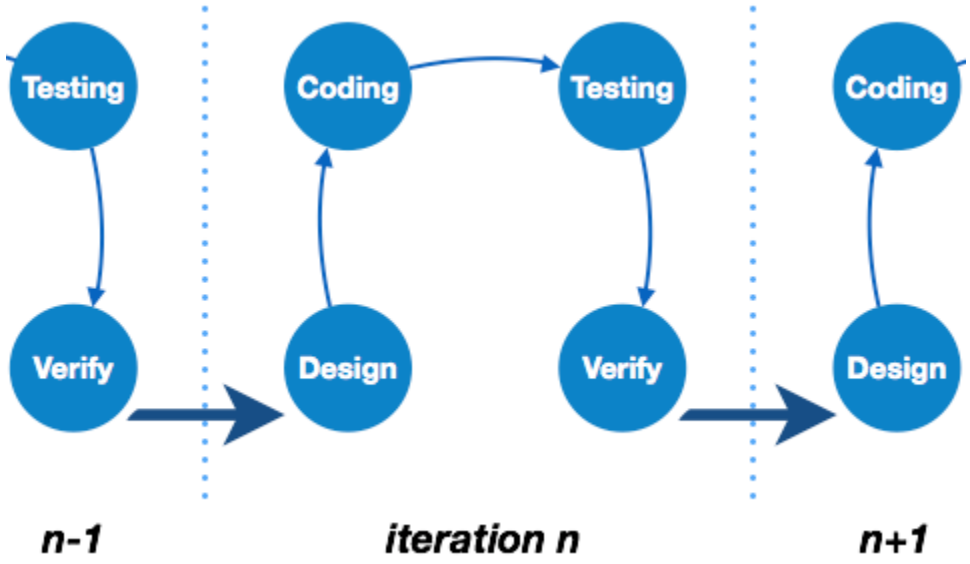
Introduction……

- Waterfall Model:
  - is the simplest model of software development paradigm.
  - It says the all the phases of SDLC will function one after another in linear manner.
  - This model is best suited when developers already have designed and developed similar software in the past and are aware of all its domains.



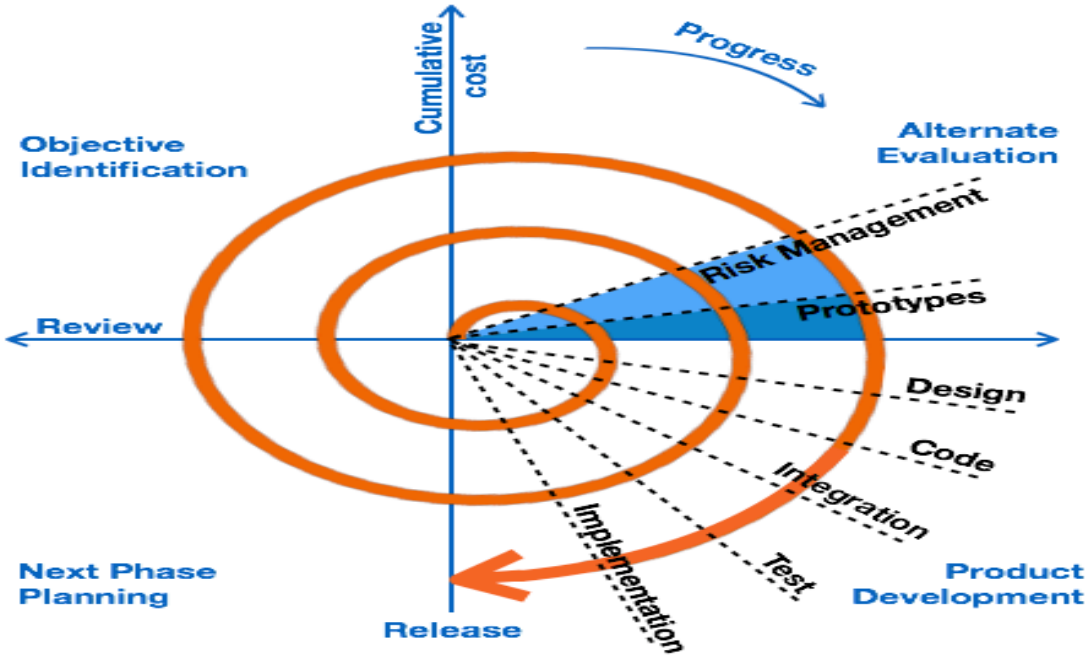
Introduction……

- Iterative Model:
  - This model leads the software development process in iterations.
  - It projects the process of development in cyclic manner repeating every step after every cycle of SDLC process.
  - Because a cycle includes small portion of whole software process, it is easier to manage the development process but it consumes more resources.



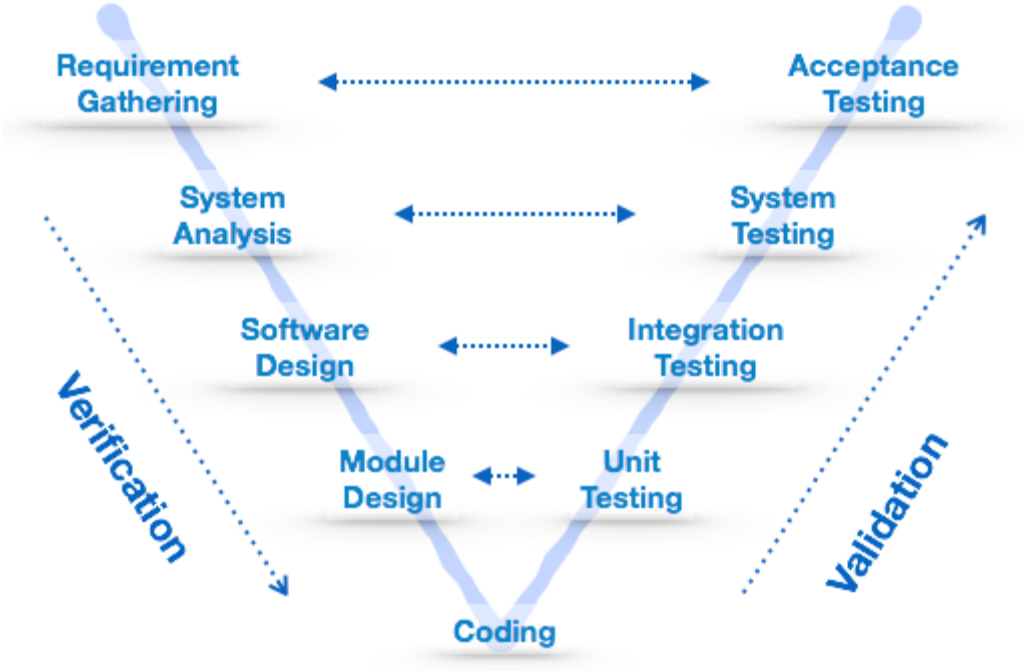
Introduction…..

- Spiral Model:
  - is a combination of both, iterative model and one of the SDLC model.
  - This model considers risk, which often goes un-noticed by most other models.
  - The model starts with determining objectives and constraints of the software at the start of one iteration.
  - Next phase is of prototyping the software. This includes risk analysis.
  - Then one standard SDLC model is used to build the software.
  - In the fourth phase of the plan of next iteration is prepared.



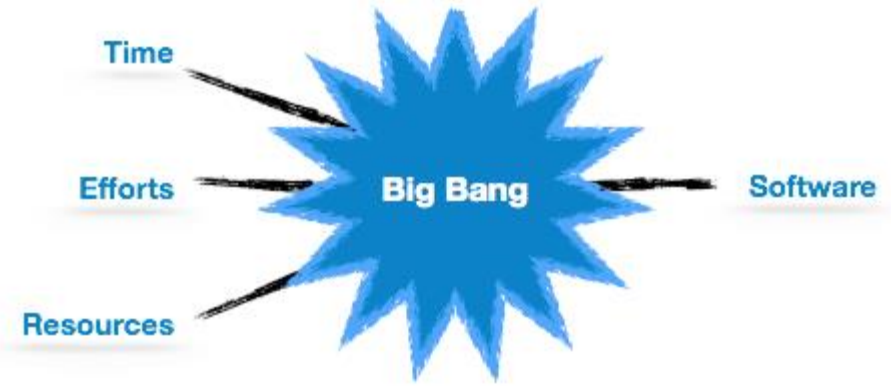
Introduction……

- V-Model:
  - The major drawback of waterfall model is we move to the next stage only when the previous one is finished and there was no chance to go back if something is found wrong in later stages.
  - V-Model provides means of testing of software at each stage in reverse manner.
  - At every stage, test plans and test cases are created to verify and validate the product according to the requirement of that stage.
  - This makes both verification and validation go in parallel.
  - This model is also known as verification and validation model.



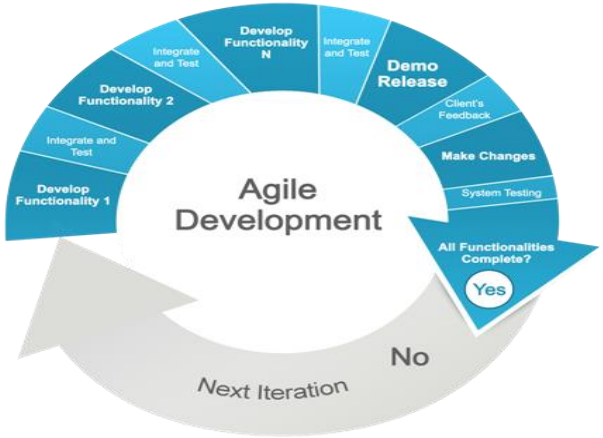
Introduction…..

- Big Bang Model:
  - This model is conceptualized around the big bang of universe.
  - As scientists say that after big bang lots of galaxies, planets and stars evolved just as an event.
  - Similarly, if we put together lots of programming and funds, you may achieve the best software product.
  - very small amount of planning is required, input requirements are arbitrary.
  - model is not suitable for large software projects but good one for learning and experimenting.



Introduction…..

- Agile Software Development:
  - often more specifically Scrum, is a different approach to software development.
  - It is more iterative and flexible when compared to the Waterfall Model.
  - Developments are done in time limited sprints, which often last 2 or 3 weeks.
  - The goal of each sprint is to deliver something production-worthy.
  - Rather than testing occurring after development, testing is concurrent to development.
  - Objectives are prioritized.
  - A sprint runs for a set period of time and delivers what it delivers, typically the most important outstanding piece.
  - Then there is a retrospective and then the next sprint begins.
  - It can happen that the product of one sprint is the work load for a following sprint.



Object Oriented Analysis and Design

- is a popular technical approach for analyzing, designing an application, system, or business by applying the object-oriented paradigm and visual modeling throughout the development life cycles to foster better stakeholder communication and product quality.
- According to the popular guide Unified Process, OOAD in modern software engineering is best conducted in an iterative and incremental way.
- Iteration by iteration, the outputs of OOAD activities, analysis models for OOA and design models for OOD respectively, will be refined and evolve continuously driven by key factors like risks and business value.

OOAD

- “Owning a hammer doesn't make one an architect"
- Knowing an object-oriented language (such as Java) is a necessary but insufficient first step to create object systems.
- Knowing how to "think in objects" is also critical.
- UML is just a standard diagramming notation. It is just a tool, not a skill that is valuable in itself.
- Knowing UML helps you communicate with others in creating software, but the real work in this course is learning Object-Oriented Analysis and Design, not how to draw diagrams.

OOAD

- The most important skill in Object-Oriented Analysis and Design is assigning responsibilities to objects. That determines how objects interact and what classes should perform what operations.
- All Software Analysis and Design is preceded by the analysis of requirements.
- One of the basic principles of good design is to defer decisions as long as possible. The more you know before you make a design decision, the more likely it will be that the decision is a good one.
- TFCL: ***Think First, Code Later!***

OOAD

- Writing Use Cases is not a specifically Object Oriented practice.
- But it is a best practice for elaborating and understanding requirements.
- A standardized approach to analysis and design helps to ensure that all necessary tasks are understood and completed in software development.
- We will focus on the Unified Process developed at Rational Software by Ivar Jacobsen, Grady Boch, Jim Rumbaugh, and others.
- Requirements Analysis, Object-Oriented Analysis and Object-Oriented Design are not a complete toolkit for a software developer.
- There are many other skills necessary in Software development, including programming.

OOAD

- Analysis
  - Analysis emphasizes an investigation of the problem and requirements, rather than a solution
  - Analysis is a board term, best qualified as requirement analysis(an investigation of the requirements) or OOA(an investigation of domain objects)
  - For e.g. if a new online trading system is desired, how will it be used? What are its functions? Comes in Analysis.
- Design
  - Design emphasizes a conceptual solution(in software and hardware) that fulfills the requirements, rather than its implementation.
  - For e.g. a description of a database schema and software objects.
  - Design can be implemented, and implementation(code) expresses true and complete realized design.
  - In nutshell , do the right thing(Analysis), and do the thing right (design).

Object Oriented Analysis and Design

- Object Oriented Paradigm:
  - The object-oriented paradigm took its shape from the initial concept of a new programming approach, while the interest in design and analysis methods came much later.
  - The first object–oriented language was Simula (Simulation of real systems) that was developed in 1960 by researchers at the Norwegian Computing Center.
  - In 1970, Alan Kay and his research group at Xerox PARK created a personal computer named Dynabook and the first pure object-oriented programming language (OOPL) - Smalltalk, for programming the Dynabook.
  - In the 1980s, Grady Booch published a paper titled Object Oriented Design that mainly presented a design for the programming language, Ada. In the ensuing editions, he extended his ideas to a complete object–oriented design method.
  - In the 1990s, Coad incorporated behavioral ideas to object-oriented methods.
- Others are Object Modelling Techniques (OMT) by James Rumbaugh and Object-Oriented Software Engineering (OOSE) by Ivar Jacobson.

OOAD

- Object Oriented Analysis:
  - OOA is the procedure of identifying software engineering requirements and developing software specifications in terms of a software system's object model, which comprises of interacting objects.
  - The main difference between object-oriented analysis and other forms of analysis is that in object-oriented approach, requirements are organized around objects, which integrate both data and functions.
  - They are modeled after real-world objects that the system interacts with.
  - In traditional analysis methodologies, the two aspects - functions and data - are considered separately.

OOAD

- *"Object-oriented analysis is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain"*. Grady Booch.
- The primary tasks in object-oriented analysis (OOA) are:
  - Identifying objects
  - Organizing the objects by creating object model diagram
  - Defining the internals of the objects, or object attributes
  - Defining the behavior of the objects, i.e., object actions
  - Describing how the objects interact
- The common models used in OOA are use cases and object models.

OOAD

- Object Oriented Design:
  - involves implementation of the conceptual model produced during object-oriented analysis.
  - In OOD, concepts in the analysis model, which are technology−independent, are mapped onto implementing classes, constraints are identified and interfaces are designed, resulting in a model for the solution domain, i.e., a detailed description of how the system is to be built on concrete technologies.
  - The implementation details generally include:
    - Restructuring the class data (if necessary),
    - Implementation of methods, i.e., internal data structures and algorithms,
    - Implementation of control, and
    - Implementation of associations.

OOAD

- *“a method of design encompassing the process of object-oriented decomposition and a notation for depicting both logical and physical as well as static and dynamic models of the system under design”*. – Grady Booch