

A large, stylized blue cloud with a thick white outline is centered in the upper half of the image. Inside the cloud, the text "OOAD" is written in a white, bold, sans-serif font. Several smaller, fainter clouds are visible in the background.

OOAD



Agenda

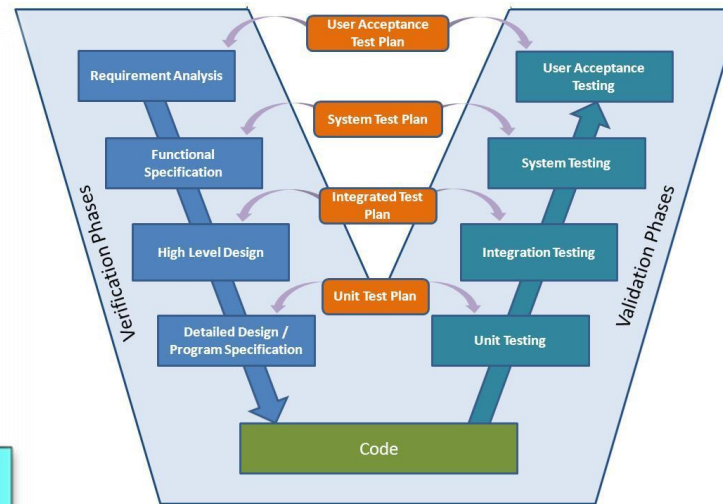
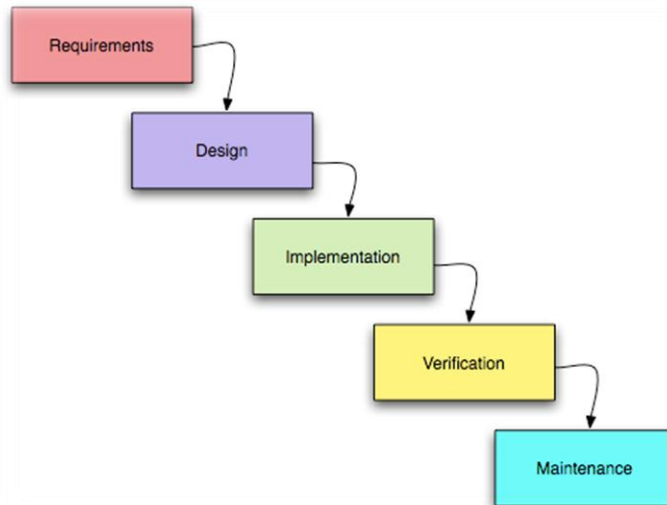


- SDLC models
- OOAD
 - UML
- Intro to OOP

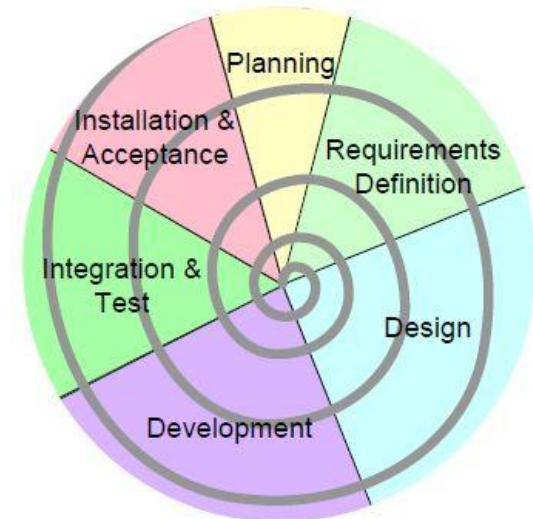
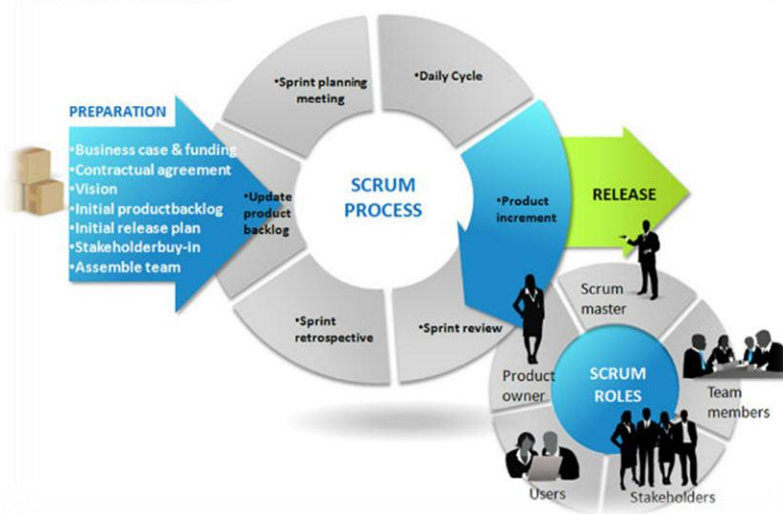


SDLC Models





SCRUM PROCESS



SDLC models



- The Waterfall Model
- The Spiral Model
- V-model
- Agile



OOAD????

- A popular technical approach for analyzing & designing an application, system, or business
- Used to create visual modeling throughout the development life cycles to foster better stakeholder communication and product quality



OO Analysis and Design



- The purpose of OO analysis and design :
 - Identifying the objects of a system.
 - Identify their relationships.
 - Make a design which can be converted to executables using OO languages.

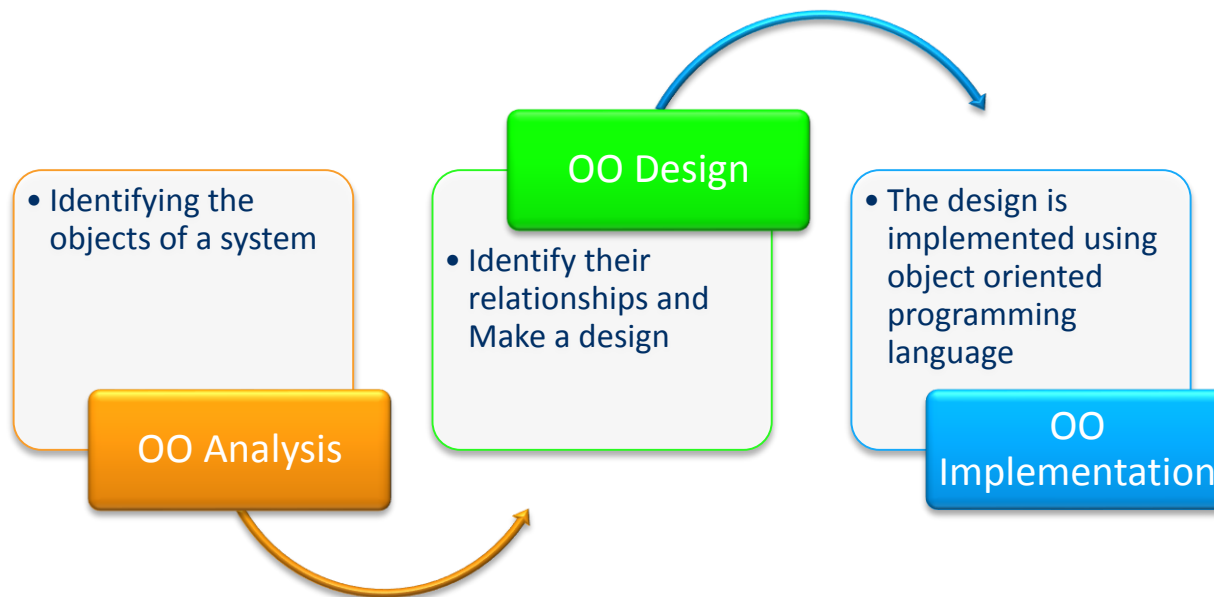


Object oriented concepts



- Fundamental concepts of object oriented world:
 - **Objects:** Objects represent an entity and the basic building block.
 - **Class:** Class is the blue print of an object.
 - **Abstraction:** Abstraction represents the behavior of an real world entity.
 - **Encapsulation:** Encapsulation is the mechanism of binding the data together and hiding them from outside world.
 - **Inheritance:** Inheritance is the mechanism of making new classes from existing one.
 - **Polymorphism:** It defines the mechanism to exists in different forms.





Object Oriented Analysis



- The primary tasks in object-oriented analysis (OOA) are:
 - Find the objects
 - Organize the objects
 - Describe how the objects interact
 - Define the behavior of the objects
 - Define the internals of the objects



Object Oriented Design



- To solve a problem that was identified and documented during object-oriented analysis
- Object-oriented design is the discipline of
 - defining the objects &
 - their interactions



Object Oriented Implementation



- The design is implemented using object oriented languages like Java, C++ etc.

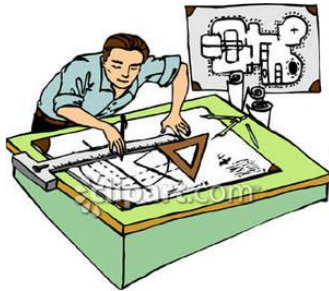


Role of UML in OO analysis & design

- UML is a modeling language used to model software and non software systems
- The OO analysis & design is transformed into UML diagrams according to the requirement



- Architect - design buildings



Blue Print



- Builders -Create buildings



History!!!



3 Amigos

Booch

Jim



Ivar

- Booch
 - **Grady Booch , Rational Corp.,**
 - Excellent for design and implementation
- OMT (Object Modelling Technique)
 - **Jim Rumbaugh, General Electrics.,**
 - Best for analysis
- OOSE (Object Oriented Software Engineering)
 - **Ivar Jacobson**
 - Use Case - Powerful technique for understanding the behavior of entire system



- The UML is a graphical language for capturing the artifacts of software developments.
- The language provides us with the notations to produce models.
- The UML is adopted industry wide language.
- The UML was originally designed by the Three Amigos at Rational Corp.



UML???

- UML stands for Unified Modeling Language.
- The language is very rich, and carries with it many aspects of Software Engineering best practice
- UML is different from the other common programming languages like C++, Java, COBOL etc.
- UML is a pictorial language used to make software blue prints.



Why UML?



- The Unified Modeling Language (UML) offers a way to visualize a system's architectural blueprints in a diagram
 - Any activities (jobs)
 - Individual components of the system
 - And how they can interact with other software components.
 - How the system will run
 - How entities interact with others (components and interfaces)
 - External user interface





Business Analyst



Designer

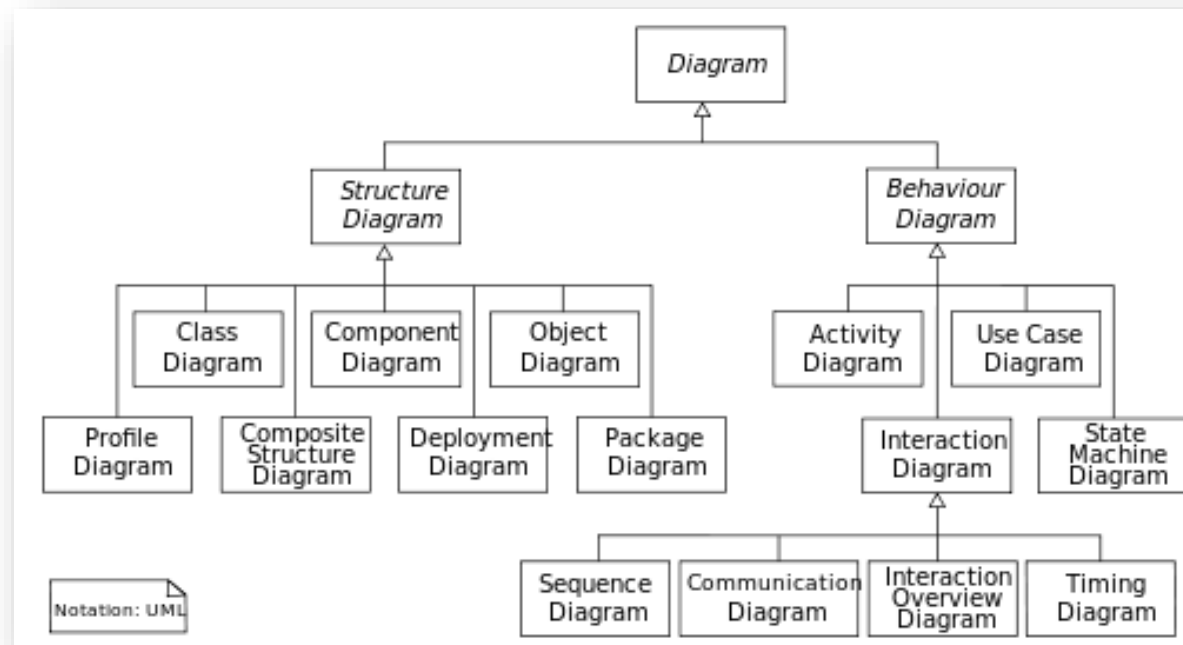


Programmer

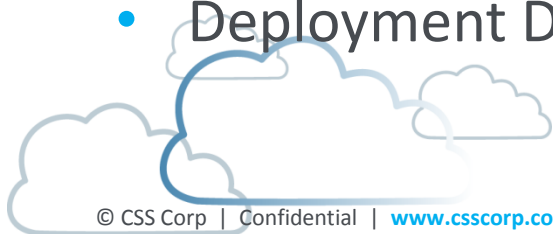


Tester

Types of Diagrams



- Use Cases - How will our system interact with the outside world?
- Class & Object Diagram - What objects do we need? How will they be related?
- Collaboration Diagram - How will the objects interact?
- Sequence Diagram - How will the objects interact(order/time)?
- State Diagram - What states should our objects be in?
- Package Diagram - How are we going to modularize our development?
- Component Diagram - How will our software components be related?
- Deployment Diagram - How will the software be deployed?



Conceptual model



A conceptual model



- *What is a conceptual model?*
 - It helps to understand the entities in the real world and how they interact with each other.
 - Conceptual Modeling (sometimes called Domain Modeling) is the activity of finding out which concepts are important to our system



- On the conceptual model, we aim to capture all of the concepts or ideas that the customer recognizes.
- For example, some good examples of concepts would be:
 - Lift in a lift control system
 - Order in a home shopping system
 - Footballer in a PlayStation football game
 - Trainer in a online e-learning system
 - Room in a room booking system

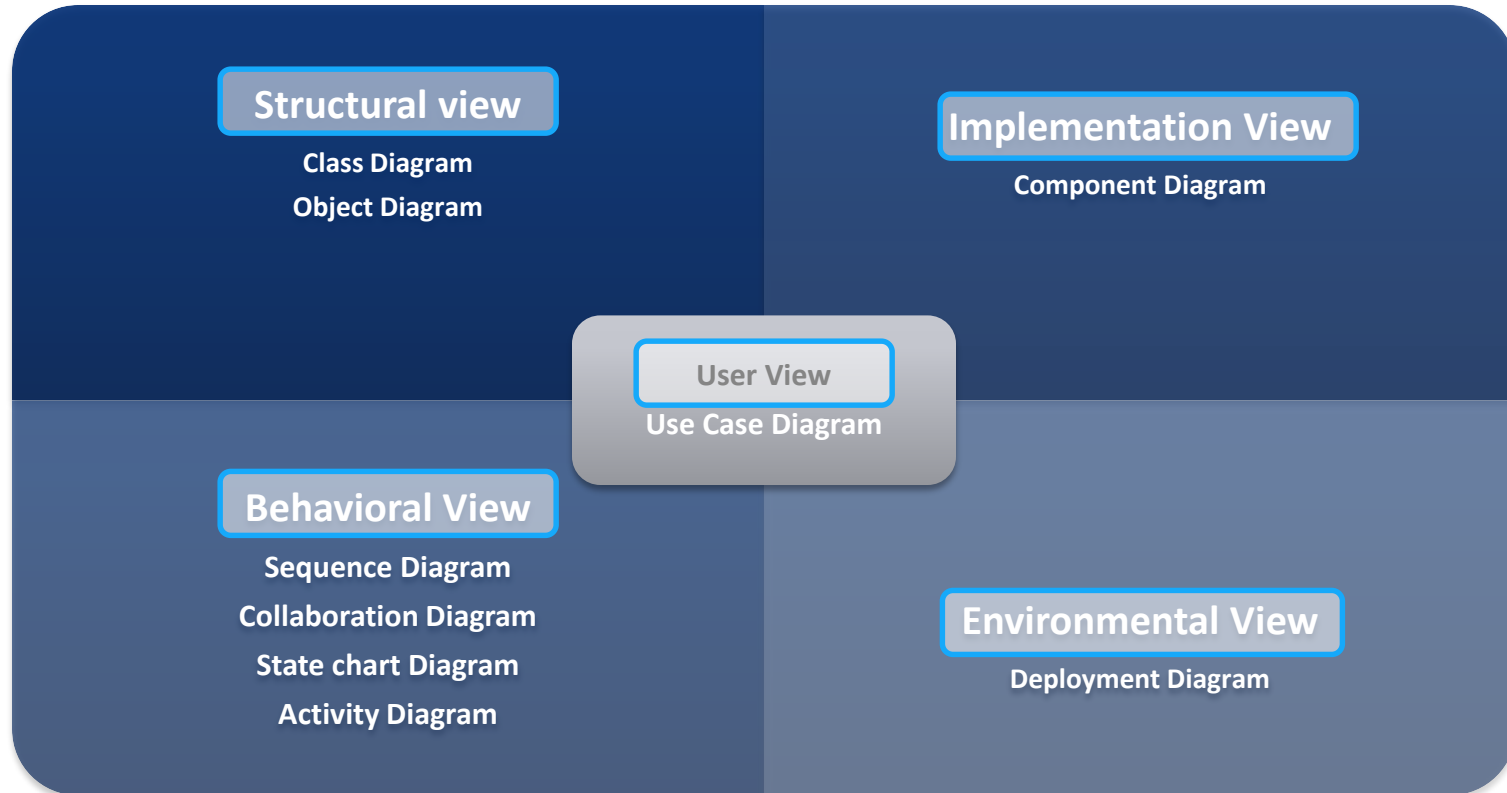


- EventTrigger - the special process that waits for 5 minutes and then tells the system to wake up and do something
- CustomerDetailsForm - the window that asks for details of the new customer in a shopping system
- DbArchiveTable - the database table holding a list of all old orders

These are bad concepts, because they are focusing on design the solution, and not the problem



Perspectives



UML Diagrams

- Use case diagrams
- Class diagrams
- Object diagrams
- Sequence diagrams
- Collaboration diagrams
- State chart diagrams
- Activity diagrams
- Component diagrams
- Deployment diagrams





Thank You

© CSS Corp

The information contained herein is subject to change without notice. All other trademarks mentioned herein are the property of their respective owners.