Unified Modeling Language UML

- Examining the Benefits of Modeling Software
  - The development team must create a series of conceptual models.
    - What is a Model?
      - A model is a simplification of reality.
      - A description of static and / or dynamic characteristics.
      - A model is an abstracts conceptualization of some entity or a system.
      - A model is abstract by its nature.
    - Model Software
      - A software model can also be represent by different diagrams views and you can create diagrams for all of these views.
      - Why model software?
we build models so that we can better understand the system we are developing.
    - Specifically, modeling enables you to.
      - -Visualize new existing systems.
-Business process.
-Use case diagram is used to capture the the business "system".
    - Communicate decisions to the project stakeholders.
      - Can be easily underload by the business owner, domain experts, and other client side stakeholders.
    - Document the decisions made in each OOSD workflow.
      - Sometimes only the project manger and the system Archived will gather requirements.
    - Specify the structure (static) and Behavior (dynamic) elements of a system.
      - Design and implementation workflows the structural and dynamic (requeriments model)
      - Solution model represents the complete conceptualization.
- Defining the UML
  - Is a graphical language for visualizing, specifying, constructing, and documentary the artifacts.
- OOSD as model transformations.
  - as the mental models.
During the requirements Gathering and Analysis, are transformed into the requirements model.
  - The functionals requirements are transformed into the requirements model.
  - The design model is merged with the arquitecture model, to procure the solution model.
  - The solution model is used to guide the construction of the code.
  - There are two broad categories of elements:
    - things (also called models)
    - relationships (also called links)
- Using the UML, a model is composed of:
  - Elements (things and relationships)
  - Diagrams (built from elements)
  - Views (diagrams showing different perspectives of a model)
- UML Diagrams
  - -Create visualizations of your mental models
-diagrams are used to construct many of the artifacts.
  - Diagram
    - Case, class, object, communication, sequence, activity, state, component, deployment, package, Interaction overview, Timing, composite Structure, profile.
    - The UML diagrams can be categorized into two main ategories:
      - Structural
        - Shows the state structure of the objects in a system
      - Behavioral
        - Show the dynamic behavior of objects in a system.
- Common UML Elements and Connectors
  - Package
    - Is used to group together any UML elements and diagrams.
Packages are elements.
Is a logical view.
  - Note
    - A note allows textual notes to be added to any aspect of a diagram.
The allows additional information in the form of text to be attached (with a dashed line).
Notes can annotate classes, methods, components, actors, associations, and so on.
  - Dependency
    - The dependency notation shows that one UML element depends on another UML element.
The type of dependency can be attached.
  - Stereotypes
    - Are used to declare a more specific type of element or Connector type.
- What UML Is and Is Not
  - UML modeling tools:
    - May prohibit you from placing an actor node in an object diagram
Restricting the drawing operations of the UML tool.
  - Support (or enforce)
    - Maintain a semantic connection
  - Generate code skeletons from the UML Diagrams.
    - -This process of generative code from models is called forward engineering.
-Some tools can also generate UML existing source code. This is called reserve engineering.
-Some tools also provide bruitt-in support for version control.
-Automatic generation of modeling elements for design patterns.
- Class Diagrams
  - Associations
    - Represent relationships between classes.
  - Multiplicity
    - Determines how many objects might participate in the relationshipñ
  - Navigation
    - Arrows on the association determine what direction an association can be traversed at runtime.
  - Associations
    - Aggregation
      - implies a relationship where the child relationship where the child can exist indeoendently of the parent.
    - Composition
      - Where the child cannot exist independent of the parent.
- Summary
  - -The OOSD Process starts with gathering the system requirements and ends with deploying a working system.
-Workflows (disciplines) define the activities that transform the artifacts.
-Use UML as your primary tool to create visual representations.

XMind : ZEN | Trial Mode