

Appendix

Appendix A

Information system : interrelated hardware and software component to support decision making, control, analysis and visualization of an organization.

Unified Modelling Language: uses graphical notation to create visual models of an OOP software system.

a general-purpose modelling language which helps to simplify the process of software design.

Standard way to specify, visualize, construct and document artefacts of a software system.

History of UML:

Main motivation behind developing the UML is to standardize the disparate notational systems and approaches to software design.

standardization of notation that seemed less elaborate rather than the standardization of methods

Booch method of Gary Booch, Object Modelling Technique of James Rumbaugh and Object-Oriented Software Engineering (OOSE) by Ivar Jacobson, published a new notation name UML version 0.9

Software development lifecycle: the series of process from requirement analysis to deployment and maintenance to meet the required objectives.

Rational Unified Process: Describe the properties like use case driven, iteration and incremental, and architecture-centric to get most out of UML

Iteration: iteration is similar to the software development life cycle phase of analysis, design, implementation, and testing.

Inception: the scopes of the project are described, and the business rationale of the project is explained.

Elaboration: plans for the construction for the project are executed by collecting more detailed requirements and doing a high level of analysis and design.

Structural Diagrams: feature the static structure of a system that convey the system concept in time independent manner and their relationship with each other.

Class diagram: static structure that describes what exists and what attributes and behavior it has, rather than how something is done.

Association: way to notate a property in a class.

Aggregation: an object of one class can own or access by the object of another class

Composition: just like an aggregation except the 'whole object' controlled the 'part object'. if a composition is destroyed, all its parts are destroyed with it.

Generalization: is a parent to child class relationships. the child inherits the attributes, operation and relationships that are defined in the parent and reuse them in one or more child model elements.

Behavioral Diagrams: dynamic behavior of a system that captures the various kinds of interaction and instantaneous state within the model.

Use case diagram: shows the functional requirements of a system describing the interaction between the users and the system

sequence diagram: describes how groups of objects collaborate in some behavior

Activity diagram: behavior diagrams that shows the flow of control or object flow with emphasis on sequence and condition of the flow.

State machine diagram: shows the discrete behavior of the designed system through finite state transitions

Object-oriented design: process to design and implement a system as a collection of interacting stateful object with specific structure and behavior.

Class and object: Class is a blueprint that describes the content of an object. An object is anything which has a state, behavior and identity that is used to create the instance of a class.

Encapsulation: used to hide the values or state of structured data object inside a class, preventing unauthorized parties with direct access to them.

Abstraction: removing irrelevant details of implementation to focus on the relevant details

Open-Closed Principle (OCP): "A module should be open for extension but closed for modification."

Liskov Substitution Principle (LSP): "Subclasses should be substitutable for their base class"

Dependency Inversion Principle (DIP): "dependent upon Abstractions. Do not depend upon concretions."

Interface Segregation Principle (ISP): "Clients should not be forced to depend upon interfaces that they do not use."

Single Responsibility Principle (SRP): "The class should have only one reason to change"

Content Management System (CMS): CMS provides support to users to create and manage the content for a web application

Web Content Management System: designed to provide capabilities for multiple users at different permission level to create, edit, publish, and report the content.

Enterprise Content Management System: ECMS is an application that supports multiple users manage, store and deliver content in a collaborative environment.

WordPress: Open source, graphical user interface feature and user-friendly content management system.

Customer: the person who uses the application to see the available services and can book and buy the listed products.

System admin: the person responsible to manage all the backend data and information and perform all the clerical role related to orders of customers.

Services: a list of available destination that interest the customer. it shows the basic information about the destination, maximum number of people who can visit, price and ability for feedback.

Booking: feature of an application for the successful purchase of the services from the customers.

Strength: Economy growth, employment opportunity, development of new project with availability of open source

Weakness: Lack of government policies, lack of security, geographical conditions

Opportunity: opportunity for entrepreneurs to establish new services and products.