UML: Unified Modeling Language

UML surge para simplificar sistemas software orientados a objetos. El modelado comenzó en los años 70 para temas de análisis y diseño.

Ventajas del UML

- Es un estándar definido a través de un metamodelo (Modelo de modelos)
- Notación gráfica fácil de aprender y usar
- Aplicable para modelar sistemas de software en diversos dominios
- Fácilmente extensible

Desventajas

- UML es un estándar, no una metodología.
- No cubre todas las necesidades de especificación de un proyecto (Por ejemplo, no cubre interfaces)
- Puede resultar complejo alcanzar un conocimiento completo del lenguaje
- Utiliza en exceso la herencia
- Faltan ejemplos elaborados

Objetivos del UML

- Busca crear una representación de nuestro software
- Busca visualizar, especificar, construir y documentar nuestro software
- Cubre todas las necesidades de especificación como recogida de requisitos, análisis, diseño, implementación y despliegue
- UML cubre toda la documentación de un sistema: Arquitectura de sistema y sus detalles, requisitos y pruebas, modelado de actividades de planificación y gestión de versiones

Conceptos de Modelado

- **Sistema**: Colección de elementos destinados a un propósito específico o general.
- **Modelo**: Simplificación de la realidad para comprender mejor el sistema
- **Vista**: Conjunto de modelos orientados a comprender una serie de funcionalidades
- **Diagrama**: Representación de los modelos

Uso de modelos

- Capturan propiedades estructurales (estáticas) y de comportamiento (dinámicas) de un sistema
- No son independientes entre sí, todos tienen cierta relación o trazabilidad
- Un modelo describe aspectos relevantes de un sistema a cierto nivel de detalle

- Los modelos definen que metodología se puede utilizar para un desarrollo

Vistas arquitecturales

Son vistas interrelacionadas que depende den los casos de uso a partir de los cuales se especifica diseño, implementación, interacción y despliegue. OJO: Las vistas no forman parte de la especificación UML.

Las 4+1 Vistas arquitecturales

- Vista de casos de uso
- Vista de diseño
- Vista de proceso
- vista de implementación
- Vista de despliegue

Estas vistas tienen varios aspectos importantes:

- Aspectos estáticos: Diagramas de casos de uso
- Aspectos dinámicos: Interacción y estado.
- UML puede implementar todos estos aspectos.

Vista de casos de uso

- Captura la funcionalidad del sistema tal y como es
- Describen la funcionalidad en base a casos de uso
- Esta vista no especifica la organización real del sistema de software

Vista de diseño

- Describe clases, interfaces y colaboraciones de nuestro sistema
- Los elementos de esta vista dan soporte a los requisitos funcionales
- Diagramas de clases y objetos

Vista de procesos

- Captura el flujo de control del software
- Requisitos no funcionales
- Diagramas de interacción, estados y actividades

Vista de implementación

- Conjunto de artefactos necesarios para la puesta en marcha
- Se usa un diagrama de componentes, de artefactos, estructura o actividades.

Vista de despliegue

- características de la ejecución del sistema
- Esta indica como instalar y ejecutar componentes de la vista de implementación

Elementos de UML

La aplicación eficaz de UML requiere conocer y comprender su metamodelado.

Bloques de construcción

Son elementos, relaciones y diagramas

- Elementos, hay 4 tipos: Estructurales, de comportamiento, de agrupamiento y de notación.
  - Estructurales: Intentan modelar las partes estáticas de un modelo (Artefactos y nodos), en UML se pueden representar: Clase, Interfaz, colaboración, caso de uso, objeto, clase activa y componente
    - Clase: Esquema que permite representar objetos con las mismas características. Se representan como: Instancia:Clase (Instancia es el objeto y clase es la clase)
    - Interfaz: Colección de operaciones que especifican un servicio que puede ser ofrecido por un componente.